Citation: J. Bermúdez-Cameo, G. López-Nicolás, J. J. Guerrero. Automatic line extraction in uncalibrated omnidirectional cameras with revolution symmetry. International Journal of Computer Vision, vol. 114, issue 1, pp. 16-37, 2015

International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

Automatic Line Extraction in Uncalibrated Omnidirectional Cameras with Revolution Symmetry

J. Bermudez-Cameo · G. Lopez-Nicolas · J.J. Guerrero

Received: date / Accepted: date

Abstract Revolution symmetry is a realistic assumption for modelling the majority of catadioptric and dioptric cameras. In central systems it can be described by a projection model based on radially symmetric distortion. In these systems straight lines are projected on curves called line-images. These curves have in general more than two degrees of freedom and their shape strongly depends on the particular camera configuration. Therefore, the existing line-extraction methods for this kind of omnidirectional cameras require the camera calibration by contrast with the perspective case where the calibration is not involved in the shape of the projected line-image. However, this drawback can be considered as an advantage because the shape of the line-images can be used for self-calibration. In this paper, we present a novel method to extract line-images in uncalibrated omnidirectional images which is valid for radially symmetric central systems. In this method we propose using the plumb-line constraint to find closed form solutions for different types of camera systems, dioptric or catadioptric. The inputs of the proposed method are points belonging to the line-images and their intensity gradient. The gradient information allows to reduce the number of points needed in the minimal solution improving the result and the robustness of the estimation. The scheme is used in a line-image extraction algorithm to obtain lines from uncalibrated omnidirectional images without any assumption about the scene. The algorithm is evaluated with synthetic and real images showing good performance. The results of this work have been implemented in an open source Matlab toolbox for evaluation and research purposes.

Keywords Omnidirectional Vision · Line Features · Uncalibrated Images · Fisheye · Catadioptric · Single-image · Matlab Toolbox

J. Bermudez-Cameo
Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain
E-mail: bermudez@unizar.es

G. Lopez-Nicolas
Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain
E-mail: gonlopez@unizar.es

J.J. Guerrero
Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain
E-mail: josechu.guerrero@unizar.es

1 Introduction

The projection of a straight line on an image is in general a curve called line-image. In conventional perspective cameras, a straight line is projected on the image via a projection plane and the resulting line-image is a 2D straight line which can be recovered from 2 image points. Notice that both the line-image and the projection plane have 2 degrees of freedom (DOF). As consequence, part of the geometry of the 3D line, which has 4 DOFs, is lost in the projection (unless in non-central systems (Gasparini and Caglioti, 2011)). In non perspective central systems the projection surface of a straight 3D line is also a plane with 2 DOFs, however the line-image is a curve with more than 2 DOFs due to the non-linearity of the projection model. The additional DOFs of these curves are deeply related with the calibration of the system which is needed to define one of these curves from 2 points. Moreover, the projection plane provides a constraint which can be exploited to estimate the calibration whenever more than two independent constraints are available (Devernay and

Faugeras, 2001; Alvarez et al, 2009; Brown, 1971). Consequently, with the appropriate approach, line-images can be extracted without calibration.

Most approaches to extract lines from omnidirectional central images use camera calibration to back project the image points to normalized rays lying on a unitary sphere around the viewpoint (Ying and Hu, 2004b; Bazin et al, 2010). In this space the representation of the line projection becomes linear and classical approaches can be used. In the particular case of central catadioptric systems the curve defining a line-image is a conic. Some approaches extract these conics directly on the images. In (Barreto and Araujo, 2005) conics are computed using classical conic fitting approaches, hence 5 DOFs are required in this fitting which is noise sensitive. In (Bermudez-Cameo et al, 2012b) conics are extracted on hypercatadioptric images although using the calibration of the camera in the fitting. In (Ying and Zha, 2005; Cucchiara et al, 2003) a Hough transform approach is used to simultaneously extract lines and calibration parameters from uncalibrated omnidirectional images.

In the case of non linear projection with unknown calibration, the location of additional edge points lying on the curve provides additional independent constraints. This kind of constraint is known as plumb-line constraint (Sturm et al, 2011) and it is one of the approaches used for self-calibration in central distorted images. Most of the plumb-line based approaches solve radial distortion models based on a transformation of the radius of an image point after a linear projection. Some of these models are the "even order polynomial model" used in (Swaminathan and Nayar, 2000; Devernay and Faugeras, 2001; Thormählen et al, 2003; Rosten and Loveland, 2011) and the " division model" proposed by Fitzgibbon in (Fitzgibbon, 2001) and extensively used in (Strand and Hayman, 2005; Wang et al, 2009; Bukhari and Dailey, 2013; Melo et al, 2013). These models are usually used to describe radial distortion on conventional cameras and they can also be used to fit catadioptric and dioptric systems.

Other kind of projection models try to describe the physical phenomenon and the behaviour of the system, e.g. the sphere camera model (Baker and Nayar, 1999; Geyer and Daniilidis, 2000, 2001) for catadioptric systems (modelling the reflection of the light on the mirror) or the classical fisheye models: the equiangular-fisheye model (also known as equidistant projection), the equisolid-fisheye model, the stereographic-fisheye model and the orthographic fisheye model (Kingslake, 1989; Stevenson and Fleck, 1996; Ray, 2002). Actually, fisheye lenses are constructed in order to satisfy such models (Sturm et al, 2011). A validation of these geometric models for real fisheyes is presented in (Schneider et al, 2009). Assuming symmetry of revolution, these models are encoded using a function h that relates the elevation angle ϕ of the projection ray with the radius of an image point r. The function $r = h(\phi)$ can be an explicit expression (Kingslake, 1989; Ray, 2002) or can be defined as a series-based expression (Tardif et al, 2006; Kannala and Brandt, 2006; Kannala et al, 2008b).

Apart from the plumb-line approach there exist other methods for calibrating omnidirectional central cameras. A survey of these methods can be found in (Puig et al, 2012). We want to remark that, in spite of existing a considerable number of methods for calibrating catadioptric systems (Wu and Hu, 2005; Barreto and Araujo, 2005; Gasparini et al, 2009; Puig et al, 2011) there are few methods for fisheye systems. A general method for central systems that can be also used for fisheyes is presented in (Scaramuzza et al, 2006). In (Mei and Rives, 2007) the sphere model is used to approximate the fisheye model. A calibration method specific for fisheyes is presented in (Kannala and Brandt, 2006; Kannala et al, 2008a).This multi-view approach uses a series-based description of the radially symmetric model. The radially symmetric series-based model is also used in a plumb-line calibration method for central systems in (Tardif et al, 2006). More recently, in (Alemán-Flores et al, 2014) line projections are detected in uncalibrated distorted images using a Hough transform approach.

In this paper we present an automatic method for line extraction in uncalibrated images which is valid for central dioptric and catadioptric cameras with symmetry of revolution. This extraction allows us to segment the collection of edges corresponding to line-images automatically. For this, we exploit two different line-image constraints: point's location (which is the constraint typically used in the plum-line approach) and point's gradient constraint, using the gradients of the intensity field. Constraints based on gradient are particularly useful when the number of DOFs of the curve is high. From each extracted line-image we obtain the corresponding projecting plane since main calibration information is implicitly computed. In our method we do not assume any restriction in the orientation of 3D lines. The input of the method is a single image containing projections of lines. The output is a set of line-images and their supporting edges, the corresponding projection planes and the main calibration parameter of the system. The main contributions of this work are the following,

– **An explicit closed form solution of the constraints developed for the catadioptric sphere camera model and for different fisheye pro-**

jection models . By contrast with (Tardif et al, 2006) each projection model is treated independently obtaining a closed solution and reducing the number of parameters to encode the distortion.

– **A gradient-based method to reduce the minimal solution**. The proposed line-image constraint is a representation of the plumb-line constraint of the image space. Working in the image space allows us to exploit the parallelism between the gradient of the intensity field and the gradient of the curve as additional independent constraint reducing the number of points needed to define a line projection when the calibration is unknown.

– **A unified parametrization for any central system with revolution symmetry**. We deal with different representations of central projections used for catadioptric and dioptric systems. Instead of calibrating all the parameters of these models we focus in a single calibration parameter $\hat{r}_{vl}$ which represents the radius of the vanishing line and it is common to all these models. Besides, we relate this parameter with the existing models and explain how to compute this relation whenever any central projection model can be expressed in the form $r = h(\phi)$.

– **A practical and robust method for automatic line-image extraction from uncalibrated omnidirectional images with revolution symmetry** . The proposed method has been implemented in a Matlab toolbox for evaluation and research purposes. This toolbox allows to extract the lines projections and the calibration of the system from a single-image.

Preliminary results of this work have been presented in (Bermudez-Cameo et al, 2013). Now, we extend the analytical solution of the plumb-line constraint directly solving the polynomials when possible. Besides, we go into detail about the gradient-base method considering different approximations. We also provide additional details and new experiments with real images. In addition, we improve the contribution with the study of the influence of the error in the principal point which was not considered in our previous work.

The rest of the paper is distributed as follows. In Section 2 we describe the covered catadioptric and fisheye projection models. In Section 3 we present the unified description to represent line-images when the system has revolution symmetry. In section 4 we present the constraints based on point's location and the constraints based on brightness gradient. Section 5 describes the algorithm to extract line-images from omnidirectional cameras without calibration. In section 6 we show the results of the experiments used to validate the method. Finally, in section 7 we show the conclusions.

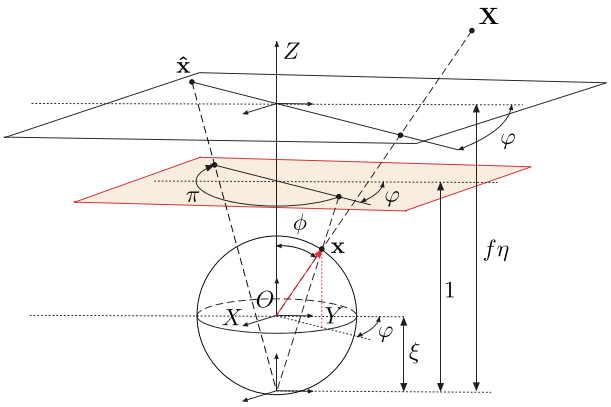
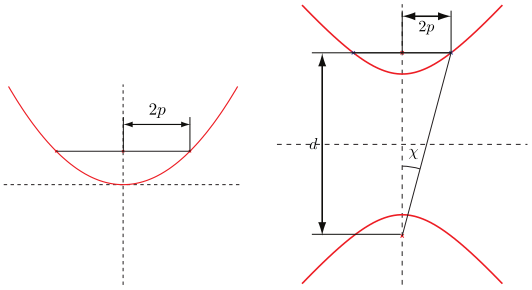


Fig. 1 Catadioptric sphere camera model: The 3D point $\mathbf{X}$ is projected onto the sphere. Then this point is backprojected to a normalized plane through a virtual optic center located a distance ξ from the effective viewpoint. This point is transformed to the point $\hat{\mathbf{x}}$ on the centred image plane using the collineation $\mathsf{H_c} = \mathsf{diag}\,(f\eta, -f\eta, 1)$.



	ξ	η
Parabolic:	1	$2p$
Hyperbolic:	$\cos\chi = \frac{d}{\sqrt{d^2+4p^2}}$	$\sin\chi = \frac{2p}{\sqrt{d^2+4p^2}}$
Planar:	0	1

Fig. 2 Parameters of the unified sphere model for catadioptric systems.

2 Projection Models for Central Systems with Revolution Symmetry

In this section we introduce the projection models covered by this work. When a projection system is central the projected rays lie on a common fixed point called viewpoint $\mathbf{O}$. Assuming the system has revolution symmetry, the reference system of the camera has the origin in the viewpoint $\mathbf{O}$ and the Z-axis is aligned with the axis of revolution. Let $\mathbf{X}$ be a 3D point in homogeneous coordinates $\mathbf{X} = (X\ Y\ Z\ 1)^{\mathsf{T}}$. The point is projected onto a unitary sphere around the viewpoint $\mathbf{O}$ of the system. It is defined with two spherical coordinates ϕ and φ as, $\mathbf{x} = (\sin\phi\cos\varphi, \sin\phi\sin\varphi, \cos\phi)^{\mathsf{T}}$ (see Fig. 1).

Depending on the projection model this point is mapped on the image using different expressions. Notice that any point lying on the revolution axis is projected on an image point called principal point. Consider the polar coordinates (r, θ) of an image point taking as reference the principal point. If the camera is correctly aligned with the axis of revolution we can observe that the coordinate θ is related with the spherical coordinate φ via the pixel aspect ratio k_{par}, as $\tan\theta = \pm k_{par} \tan\varphi$ (the sign in this expression is used to model reflections in catadioptric systems). Catadioptric and dioptric systems are projection systems that conserve the revolution symmetry. This means that the radius of a pixel $\hat{r}$ can be expressed in terms of the elevation angle ϕ. Depending on the device a different expression for $\hat{r} = h(\phi)$ is used.

In the following descriptions we assume that image points are expressed in a reference centred in the principal point. We also assume that pixel aspect ratio is equal to one which is valid in digital imagery. A point in this reference system is denominated with the notation $\hat{\mathbf{x}}$. The transformation from this reference to the final image coordinate system is the following,

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & s & u_0 \\ 0 & k_{par} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \hat{\mathbf{x}} \ . \tag{1}$$

2.1 Projection Models for Catadioptric Systems

Under the sphere camera model (Geyer and Daniilidis, 2000; Baker and Nayar, 1999; Geyer and Daniilidis, 2001) all central catadioptric systems can be modelled by a projection to the unitary sphere followed by a perspective projection via a virtual viewpoint located a distance ξ from the effective viewpoint (see Fig. 1). Let $\hat{\mathbf{x}} = (\hat{x}, \hat{y}, 1)^T$ be a point on an image referred to the principal point and given the spherical coordinates ϕ and φ of the corresponding point on the unitary sphere then,

$$\hat{x} = \frac{f\eta \sin\phi \cos\varphi}{\cos\phi + \xi} \quad \text{and} \quad \hat{y} = -\frac{f\eta \sin\phi \sin\varphi}{\cos\phi + \xi} \ . \tag{2}$$

In polar coordinates the point is described by $\hat{\theta} = -\varphi$ and

$$\hat{r} = \frac{f\eta \sin\phi}{\cos\phi + \xi} = \frac{f\eta \tan\phi}{1 + \xi\sqrt{\tan^2\phi + 1}} \ . \tag{3}$$

The geometry of the projection system is described by parameters ξ and η which have a different definition depending on the system type (see Fig. 2).

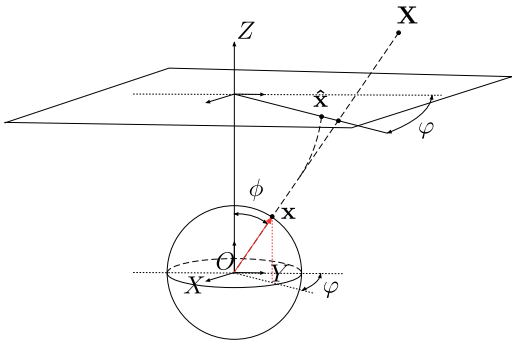


Fig. 3 Fisheye camera models: The radius of the point on the image is distorted by a function $\hat{r} = h(\phi)$.

Equiangular Fisheye	Stereographic Fisheye	Orthogonal Fisheye	Equisolid Fisheye
$f\phi$	$2f\tan\left(\frac{\phi}{2}\right)$	$f\sin(\phi)$	$2f\sin\left(\frac{\phi}{2}\right)$

Table 1 Function $\hat{r} = h(\phi)$ describing fisheye projection models.

2.2 Fisheye Models

Several models are used to describe point projection in dioptric systems depending on the manufacturing procedure of the lens (Kingslake, 1989; Stevenson and Fleck, 1996; Ray, 2002). Assuming square pixel, these models are expressed in polar coordinates $(\hat{r}, \hat{\theta})$. For all these models $\hat{\theta} = \varphi$ and the radius changes depending on the camera type (see Table 1). The parameter f of these models does not represent the focal distance of the camera like in the case of the parameter f of the sphere camera model presented in the previous section.

Notice that the stereographic projection is equivalent to the projection in a paracatadioptric system. The stereographic projection is also equivalent to the "division model" model of Fitzgibbon (Fitzgibbon, 2001) when using a single parameter ($r' = k_1 \frac{r}{1-k_2^2}$).

3 Unified Description for Line Projection in Central Systems with Revolution Symmetry

In this section we present a unified description for line-images in dioptric and catadioptric imagery (Bermudez-Cameo et al, 2012a), which is valid for central systems with revolution symmetry.

Let $\mathbf{\Pi} = (n_x, n_y, n_z, 0)^\mathsf{T}$ be a plane defined by a 3D line and the viewpoint of the system $\mathbf{O}$. The projected line associated to the 3D line can be represented by $\mathbf{n} = (n_x, n_y, n_z)^\mathsf{T}$. Then, the points $\mathbf{X}$ lying in the 3D line are projected to points $\mathbf{x}$. These points satisfy $\mathbf{n}^\mathsf{T}\mathbf{x} = 0$. Using the spherical representation and assuming that

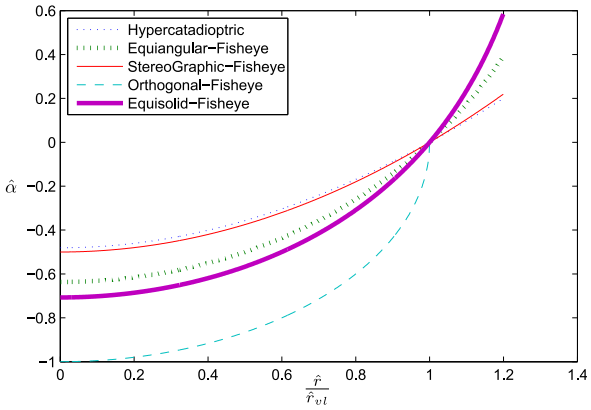


Fig. 4 Representation of $\hat{\alpha}$ depending on $\hat{r}$ for different central projection systems with revolution symmetry. $\hat{r}$ is normalized with respect to the radius of the projection of the vanishing line $\hat{r}_{vl}$.

	$\hat{\alpha}$	$\hat{r}_{vl}$	polynomial
Perspective	f	∞	straight
Para catadioptric	$\frac{\hat{r}^2}{4fp} - fp$	$2fp$	conic
Hyper catadioptric	$\frac{-f+\cos\chi\sqrt{\hat{r}^2+f^2}}{\sin\chi}$	$f\tan\chi$	conic
Equiangular-Fisheye	$-\hat{r}\cot\frac{\hat{r}}{f}$	$f\frac{\pi}{2}$	non polynomial
Stereographic-Fisheye	$\frac{\hat{r}^2}{4f} - f$	$2f$	conic
Orthogonal-Fisheye	$-\sqrt{f^2-\hat{r}^2}$	f	conic
Equisolid-Fisheye	$\frac{2\hat{r}^2-f^2}{2\sqrt{f^2-\hat{r}^2}}$	$f\frac{\sqrt{2}}{2}$	quartic

Table 2 Parameters for different central projection systems with revolution symmetry.

$\hat{\theta} = \pm\varphi$ (square pixel) this equality could be expressed as

$$\sin\phi\,(n_x\hat{x} \pm n_y\hat{y}) + n_z\hat{r}\cos\phi = 0\ . \tag{4}$$

With the change of variable $\hat{\alpha} = \dfrac{n_x\hat{x} \pm n_y\hat{y}}{n_z}$ we can isolate the model parameters from the normal describing the line, obtaining the expression,

$$\hat{\alpha} = -\hat{r}\cot\phi\ . \tag{5}$$

Notice that $\hat{\alpha} = \hat{\alpha}\,(\hat{r})$, as a result of $\phi = h^{-1}\,(\hat{r})$ when we have symmetry of revolution and square pixel. Therefore, the constraint for points on the line projection in image coordinates for systems with symmetry of revolution is

$$n_x\hat{x} \pm n_y\hat{y} - n_z\hat{\alpha}\,(\hat{r}) = 0\ , \tag{6}$$

where $\hat{\alpha}$ is a different expression for each camera model depending on the radius and the model parameters (see Table 2).

3.1 Line-Image Curve Representation and Unified Main Calibration Parameter

Equation (6) is the homogeneous representation of the line projection on the image. There exist two particular cases common to all the projection models showed above. First we have the case in which 3D lines are coplanar to the revolution axis. In this case $n_z = 0$ and the resulting line-image is a radial straight line passing through the principal point, described as

$$n_x\hat{x} \pm n_y\hat{y} = 0\ . \tag{7}$$

The second particular case happens when $\mathbf{n} = (0,0,1)^{\mathsf{T}}$. In this case the line-image is the projection of the vanishing line. This projection is a circle centred at principal point and with radius $\hat{r}_{vl}$. This radius depends only on the system geometry (see Table 2) and can be used as a main calibration parameter independently of the camera system. In this case, the line-image equation has the form,

$$\hat{\alpha}\,(\hat{r}_{vl}) = 0\ . \tag{8}$$

In Fig. 4 we show a comparison among different $\hat{\alpha}\,(\hat{r})$ functions for different central projection systems. The radius $\hat{r}$ has been normalized with respect to $\hat{r}_{vl}$. For the case of the orthogonal system, $\hat{\alpha}$ only makes sense when $\hat{r} < \hat{r}_{vl}$.

The general form for a line-image is a curve. In most cases, these curves can be also expressed as polynomials. The catadioptric case has been deeply studied in (Barreto and Araujo, 2005), and it has been proven that the line-image is a conic. The stereographic case is equivalent to the paracatadioptric projection therefore the corresponding line-image is a conic encoded with the parameters of the sphere model. The orthogonal-fisheye line-image is also a conic but it is not encoded with the parametrization of the sphere model. The equisolid line-image is a quartic (see Appendix A). For other cases in general the curve is not a polynomial.

Each line-image is the projection of any 3D line contained in a projection plane which is described by

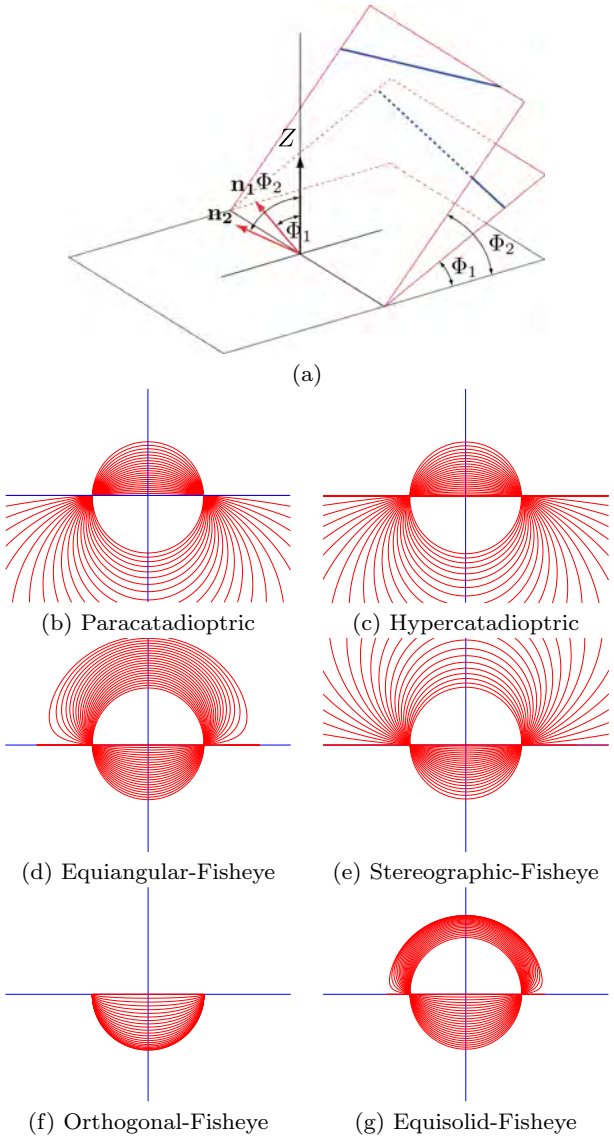


Fig. 5 Representation of line-images on the image plane depending on the projection model and with different values of the elevation angle Φ of the normal $\mathbf{n}$ representing the projection plane. (a) Elevation of the projection plane. (b-g) Line-images when increasing the elevation angle Φ for different projection models.

the normal vector $\mathbf{n}$. Consider the orientation of the central system is aligned with the vertical axis Z (see Fig. 5 (a)) and the parametrization of this normal is $\mathbf{n} = (\cos\Phi\cos\Theta, \cos\Phi\sin\Theta, \sin\Phi)^{\mathsf{T}}$ with the elevation angle Φ, and the azimuth angle Θ (do not confuse with the angles of the projecting ray of a point ϕ and φ). Given a fixed value of Φ the relation between two different line-images with different azimuth angle Θ is a rotation around the principal point. However, given a fixed value of Θ a variation in the elevation angle Φ implies a change in the curvature of the line-image. In Fig. 5, we show a parametric representation of line-images

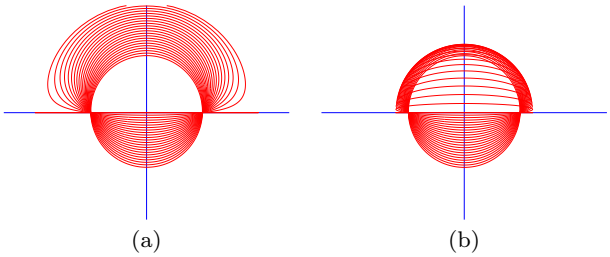


Fig. 6 Comparison of line-images on the image plane with an equiangular-fisheye system using (a) the equiangular-fisheye projection model and (b) the catadioptric sphere model with $\xi > 1$. Each line-image corresponds to a different value of the elevation of the normal $\mathbf{n}$.

for different central projection systems with revolution symmetry. We can imagine the projected line-image, as the projection of the intersection of the projecting plane with any sphere centred on the reference system. We represent different line-images with a fixed value of Θ but increasing the elevation angle Φ from 0 to $\frac{\pi}{2}$. (see Fig. 5 (a)). To compare the different systems each image has been simulated for a different model but with the same $\hat{r}_{vl}$. The outside region of this circle corresponds to a FOV greater than 180 deg and the inside with a FOV lesser than 180 deg. As in the example the system is oriented from bottom to top the inside of the circle corresponds to points with $Z > 0$ and the outside the circle corresponds with points with $Z < 0$. Notice that any line-image pass through both regions, as the projection plane intersects the both regions in the 3D space. This parametric representation corresponds to the projection plane covering 360 deg around the axis of revolution. An infinity 3D segment only fulfils a part of the curve covering 180 deg.

3.2 The Sphere Catadioptric Model as Fisheye Model

Some authors have used the catadioptric sphere model to calibrate fisheye models (Ying and Hu, 2004a; Courbon et al, 2007). The stereographic projection is equivalent to the paracatadioptric projection and it can be directly encoded using the sphere model. From the definition of Table 1 and using the half-angle formula,

$$\hat{r} = 2f\tan\left(\frac{\phi}{2}\right) = 2f\frac{\sin\phi}{\cos\phi + 1} \tag{9}$$

which is the equation (3) with $\xi = 1$ and $\eta = 2$. For other cases (Ying and Hu, 2004a; Courbon et al, 2007) propose to use the sphere model with $\xi > 1$ (in catadioptric systems where $0 < \xi < 1$).

In Fig. 6 we show a comparison of the line-images of an equiangular-fisheye system using the equiangular-fisheye projection model and the catadioptric sphere

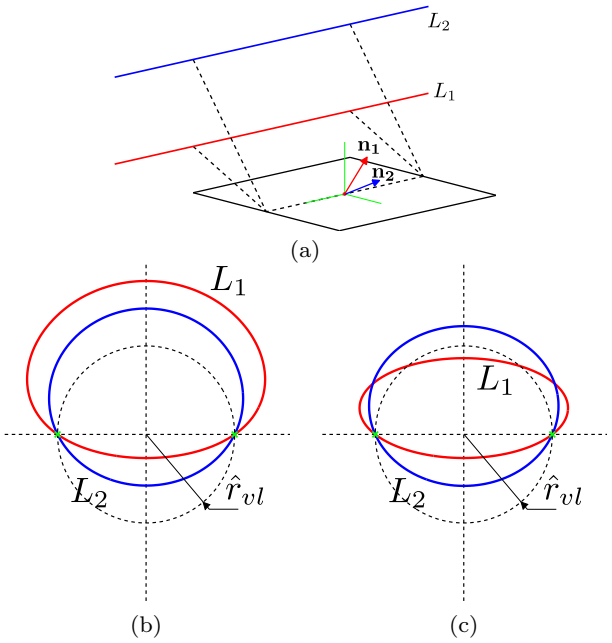


Fig. 7 (a) Isometric view of two parallel lines. Comparison of line-images of two parallel lines with an equiangular-fisheye system using the equiangular-fisheye projection model (b) and using the catadioptric sphere model (c). In green the vanishing points of direction X.

model showing the limitations of using the catadioptric sphere model for fisheyes. In order to illustrate these limitations in more detail, in Fig. 7 we represent the line projection of two parallel lines. In Fig. 7 (b) we show the line projection of these lines using the equiangular-fisheye projection model. The dotted circle represents the projection of the vanishing line. The region inside this circle corresponds to a FOV lesser than 180 deg and the projections of 3D points with $Z > 0$. The region outside this circle corresponds to a FOV greater than 180 deg and the projections of 3D points with $Z < 0$. Consider the line-image L1. The part of L1 located inside the circle corresponds with the part of the plane with $Z > 0$ and the part of L1 located outside the circle corresponds with the part of the plane with $Z < 0$. As the 3D lines are parallel the line-images intersects in two points corresponding with the vanishing points (in green) of the direction of the lines. One of the points corresponds to the positive direction and the other with the negative. In Fig. 7 (c) we show the projection of these lines using the catadioptric sphere model with $\xi > 1$ proposed in (Ying and Hu, 2004a; Courbon et al, 2007). Consider the line-image L1. The part of the plane with $Z > 0$ is projected inside the circle, however the part of the plane with $Z < 0$ is projected in both regions: inside and outside the circle. That means that the line-image does not correctly fit the line projection when the FOV is greater than 180 deg. In addition, the line-images intersect in four points instead of two giving a sense of non-geometric coherence: two of the intersections are the vanishing points and two are points without geometric sense.

3.3 The Homogeneous Line-image Equation as a Measure of Distance

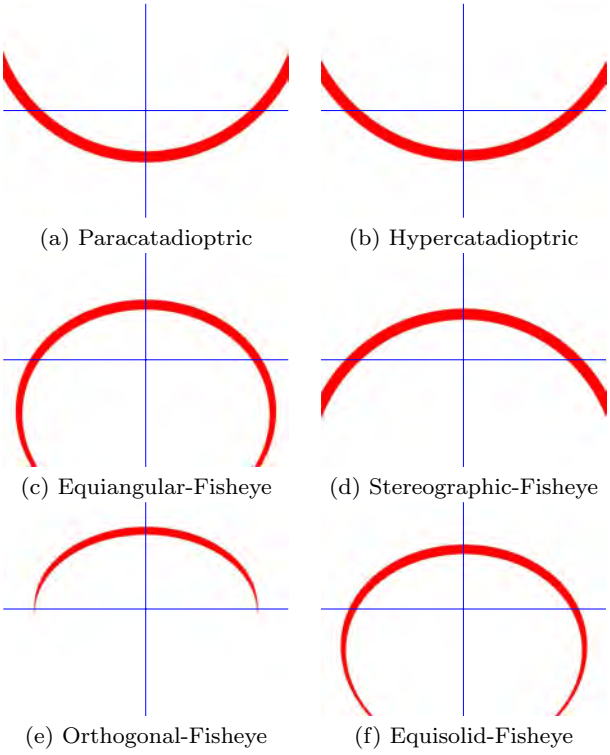
The homogeneous expression of the line-image (6) defines a family of curves located to an algebraic distance from the original curve.

$$d(\hat{x}, \hat{y}) = n_x\hat{x} \pm n_y\hat{y} - n_z\hat{\alpha} \ . \tag{10}$$

This algebraic distance is an approximation of the metric distance from a point to the line-image and is defined in pixels. In Fig. 8 we show the region defined by a fixed threshold of 20 pixels around a given line-image for different systems. The thickness of this region is homogeneous for catadioptric and stereographic systems but it is not completely homogeneous for equiangular, orthogonal and equisolid systems. However, in general this distance can be used to discriminate if a point belongs to a line-image.

Consider for example the algebraic distance based on polynomials (e.g. for hypercatadioptric systems $d = \sqrt{\mathbf{x}^\mathsf{T}\Omega_{cata}\mathbf{x}}$, see Appendix A). Given a fixed threshold, the region around the conic has a different thickness depending on the elevation angle of the vector $\mathbf{n}$. With our proposal the distance is a good approximation in regions close to the line-image.

In Fig. 9 (a) we show a comparison between the distance of a point to the line-image (blue dotted) and the proposed algebraic distance (red) for hypercatadioptric images. The algebraic distance approximates the real distance in regions which are close to the line-image, therefore can be used to discriminate if a point lies on a line-image or not. In Fig. 9 (b) we show the same comparison but using the algebraic distance defined by the expression of a conic on the image ($d = \sqrt{\mathbf{x}^\mathsf{T}\Omega_{cata}\mathbf{x}}$). We can see how this distance does not approximate well the metric distance in regions close to the curve. We also show that this distance is lower than the metric distance in vertical lines but higher when the lines are horizontal. In practice that means that the thickness of a region defined by a threshold varies considerably if elevation of $\mathbf{n}$ changes.

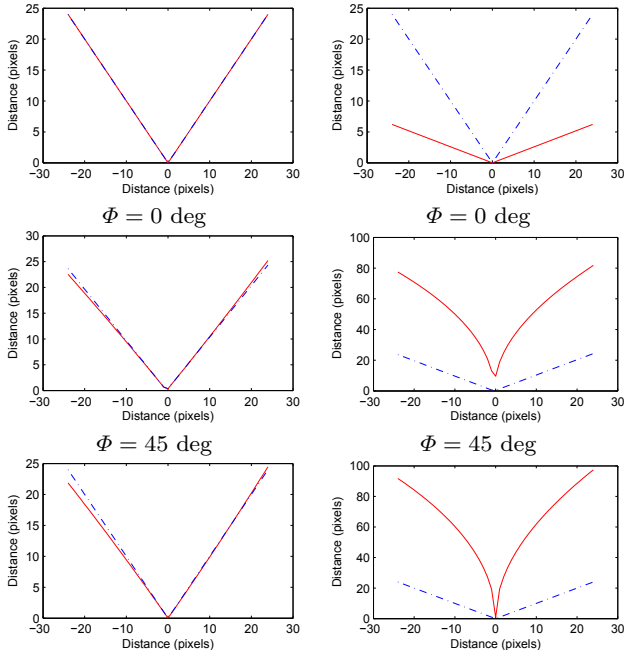


(a) Paracatadioptric (b) Hypercatadioptric

(c) Equiangular-Fisheye (d) Stereographic-Fisheye

(e) Orthogonal-Fisheye (f) Equisolid-Fisheye

Fig. 8 Region around a line-image defined by a fixed distance of 20 pixels.



(a) (b)

Fig. 9 Comparison between metric distance (blue dotted) and algebraic distances (red solid): (a) Our proposal (10). (b) Conic based algebraic distance.

3.4 Line-Image Definition from Two Points

Given at least two points lying on a line-image described by (6) we can obtain the normal $\mathbf{n}$ by solving the homogeneous linear system

$$\left(\hat{x}_i \pm\hat{y}_i -\hat{\alpha}_i\right)\mathbf{n} = 0 \quad \text{for} \quad i = 1, ..., n \quad \text{with} \quad n \geq 2. \tag{11}$$

where $\hat{\alpha}$ is a different expression for each camera system (see Table 2) and the sign '$\pm$' is positive for dioptric systems and negative for catadioptrics. The system is solved by using a Singular Value Decomposition. In particular for the minimal case of two points and solving for $\mathbf{n}$ we have

$$\mathbf{n} = \begin{pmatrix} \hat{y}_1\hat{\alpha}_2 - \hat{y}_2\hat{\alpha}_1 \\ \pm\left(\hat{x}_2\hat{\alpha}_1 - \hat{x}_1\hat{\alpha}_2\right) \\ \hat{x}_2\hat{y}_1 - \hat{x}_1\hat{y}_2 \end{pmatrix} . \tag{12}$$

4 The Straight-Line Constraint on the Image

In Section 3 we have presented an unified line-image description of the line projection in systems with revolution symmetry. We have also presented the unified main calibration parameter $\hat{r}_{vl}$ encoding the calibration of these systems. In this section we use this description to develop and to particularize the plumb-line constraint for the different projection models presented in Section 2. The goal is computing both, the main calibration parameter $\hat{r}_{vl}$ and the projecting plane of the line $\mathbf{n}$. The results are two kind of constraints: location based constraints which allow to compute the line-image and $\hat{r}_{vl}$ from the coordinates of at least three points and gradient based constraints which allow to compute the line-image and $\hat{r}_{vl}$ from the coordinates and the gradients of a minimum of two points.

4.1 Location Based Line-image Constraint

As presented in Section 3, the projecting plane Π describing the line projection is completely defined by two points and the view-point. That implies that the calibration of the system is embedded in the geometry of this curve. If the calibration of the system is known, the projecting plane normal $\mathbf{n}$ can be recovered from two image points (11). However, when constraint (6) is satisfied the line-image can be recovered even if camera calibration is unknown. Notice that when the number of equations defining (11) is greater than two we obtain a redundant system. Therefore when having three

(a) Planar	$(l_1 + l_2 + l_3) = 0$
(b) Paracatadioptric	$\hat{r}_{vl} = \sqrt{\frac{l_1\hat{r}_1^2 + l_2\hat{r}_2^2 + l_3\hat{r}_3^2}{l_1+l_2+l_3)}}$
(c) Hypercatadioptric	$\hat{r}_{vl} = \sqrt{\left(\frac{l_1\sqrt{\hat{r}_1^2+f^2} + l_2\sqrt{\hat{r}_2^2+f^2} + l_3\sqrt{\hat{r}_3^2+f^2}}{l_1+l_2+l_3}\right)^2 - f^2}$
(d) Equiangular-Fisheye	$l_1\hat{r}_1 \cot\left(\frac{\pi}{2}\frac{\hat{r}_1}{\hat{r}_{vl}}\right) + l_2\hat{r}_2 \cot\left(\frac{\pi}{2}\frac{\hat{r}_2}{\hat{r}_{vl}}\right) + l_3\hat{r}_3 \cot\left(\frac{\pi}{2}\frac{\hat{r}_3}{\hat{r}_{vl}}\right) = 0$
(e) Stereographic-Fisheye	$\hat{r}_{vl} = \sqrt{\frac{l_1\hat{r}_1^2 + l_2\hat{r}_2^2 + l_3\hat{r}_3^2}{l_1+l_2+l_3}}$
(f) Orthogonal-Fisheye	$l_1\sqrt{\hat{r}_{vl}^2 - \hat{r}_1^2} + l_2\sqrt{\hat{r}_{vl}^2 - \hat{r}_2^2} + l_3\sqrt{\hat{r}_{vl}^2 - \hat{r}_3^2} = 0$
(g) Equisolid-Fisheye	$l_1\frac{\hat{r}_1^2 - \hat{r}_{vl}^2}{\sqrt{2\hat{r}_{vl}^2 - \hat{r}_1^2}} + l_2\frac{\hat{r}_2^2 - \hat{r}_{vl}^2}{\sqrt{2\hat{r}_{vl}^2 - \hat{r}_2^2}} + l_3\frac{\hat{r}_3^2 - \hat{r}_{vl}^2}{\sqrt{2\hat{r}_{vl}^2 - \hat{r}_3^2}} = 0$

Table 3 Three points line-image constraint for different central projection systems with revolution symmetry.

points lying on a line-image the rank of the homogeneous matrix must be two. Imposing this condition we find a constraint using three points of a line image. The line-image constraint can be written as

$$l_1\hat{\alpha}_1 + l_2\hat{\alpha}_2 + l_3\hat{\alpha}_3 = 0, \quad \text{where} \tag{13}$$

$l_1 = \hat{x}_2\hat{y}_3 - \hat{x}_3\hat{y}_2$, $l_2 = \hat{x}_3\hat{y}_1 - \hat{x}_1\hat{y}_3$ and $l_3 = \hat{x}_1\hat{y}_2 - \hat{x}_2\hat{y}_1$.

Thus, with three points lying on the line-image and equation (13) it is possible to obtain the main calibration parameter $\hat{r}_{vl}$. Substituting the expression of $\hat{\alpha}$ in In Table 3 we show the corresponding expressions for each system which are computed by substituting the expression of $\hat{\alpha}$ in (13) and expressing them in terms of $\hat{r}_{vl}$.

- **Computation of $\hat{r}_{vl}$ in perspective systems.** It is not possible to extract any additional information (Table 3 (a)).
- **Computation of $\hat{r}_{vl}$ in paracatadioptric and stereographic systems systems.** In these cases the radius $\hat{r}_{vl}$ is directly computed (Table 3 (b)(e)). Notice that, as $\hat{r}_{vl}$ must be positive only the positive solution of the root square is a valid solution.
- **Computation of $\hat{r}_{vl}$ in equiangular-fisheye systems.** For equiangular-fisheye systems the constraint (Table 3 (d)) is solved by minimization. To initialize this minimization the value of $\hat{r}_{vl}$ is computed from a linear approximation of the constraint

$$\hat{r}_{vl} \approx \frac{\pi}{2}\sqrt{\frac{1}{3}\frac{(l_1\hat{r}_1^2 + l_2\hat{r}_2^2 + l_3\hat{r}_3^2)}{(l_1 + l_2 + l_3)}} \ . \tag{14}$$

  In Fig. 10 (a) we show some examples of line-images defined with 3 points in equiangular-fisheye systems.

- **Computation of $\hat{r}_{vl}$ in orthogonal-fisheye systems.** The equation in table 3 (f) becomes to the bi-quadratic polynomial equation

$$c_1\hat{r}_{vl}^4 + 2c_2\hat{r}_{vl}^2 + c_3 = 0 \tag{15}$$

  where

$$c_1 = \prod_{i=0}^{3}\sum_{j=1}^{3}\left((-1)^{\delta_{i,j}+1}\, l_j\right) \tag{16}$$

$$c_2 = \sum_{i=1}^{3} l_i^2\hat{r}_i^2 \sum_{j=1}^{3}\left((-1)^{\delta_{i,j}}\, l_j^2\right) \tag{17}$$

$$c_3 = \prod_{i=0}^{3}\sum_{j=1}^{3}\left((-1)^{\delta_{i,j}+1}\, l_j\hat{r}_j\right) \tag{18}$$

  $\delta_{i,j}$ being the Kronecker delta.
  This equation has direct solution for $\hat{r}_{vl}$. Notice that despite the bi-quadratic equation has a priori four solutions, two of them are always negative ($\hat{r}_{vl} = \pm\sqrt{\hat{r}_{vl}^2}$) and in practice the wrong one can be discarded using the equation of Table 3 (f) and rejecting the non-real solutions.
- **Computation of $\hat{r}_{vl}$ in equisolid-fisheye systems.** In equisolid-fisheye systems the equation in table 3 (g) becomes to the bi-eight degree equation

$$\sum_{m=0}^{8}\omega_m\hat{r}_{vl}^{2m} = 0 \ . \tag{19}$$

  This equation (see Appendix C) has also a direct solution for $\hat{r}_{vl}$.

Perspective	Para-Catadioptric	Hyper-Catadioptric	Equiangular-Fisheye	Stereographic-Fisheye	Orthogonal-Fisheye	Equisolid-Fisheye
0	$\frac{1}{2fp}$	$\frac{\cot\chi}{\sqrt{\hat{r}^2+f^2}}$	$\frac{1}{f}\left(1-\frac{f}{\hat{r}}\cot\frac{\hat{r}}{f}+\cot^2\frac{\hat{r}}{f}\right)$	$\frac{1}{2f}$	$\frac{1}{\sqrt{f^2-\hat{r}^2}}$	$\frac{\hat{r}\left(3f^2-2\hat{r}^2\right)}{2\left(f^2-\hat{r}^2\right)^{3/2}}$

Table 4 $\frac{\partial\hat{\alpha}}{\partial\hat{r}}\frac{1}{\hat{r}}$ for different central projection systems with revolution symmetry.

– **Computation of $\hat{r}_{vl}$ in hypercatadioptric systems (Table 3 (c)).**
In hypercatadiopric systems the focal distance of the camera is needed to compute the main calibration parameter $\hat{r}_{vl}$. This is because two calibration parameters (f and χ) are involved and they cannot be coupled to reduce a degree of freedom.
When f is known the equation in Table 3 (d) is solved for χ and it yields

$$\cos\chi=\frac{f\left(l_1+l_2+l_3\right)}{l_1\sqrt{\hat{r}_1^2+f^2}+l_2\sqrt{\hat{r}_2^2+f^2}+l_3\sqrt{\hat{r}_3^2+f^2}}\,. \tag{20}$$

The focal distance f can be directly computed using gradient constraints (see Appendix B) although the method is quite unstable. In practice it is possible to complete the extraction of the line-image with a rough approximation of f.

4.2 Gradient-Based Line-Image Constraint

Here we present the line-image gradients for central systems with revolution symmetry, which gives additional constraints for line-image fitting. They allow to reduce the minimum number of points from three to two.

On the one hand, the normal direction to a line image is described by the gradient of the homogeneous expression (10) which is expressed by

$$\frac{\partial d}{\partial\hat{x}}=n_x-n_z\frac{\partial\hat{\alpha}}{\partial\hat{r}}\frac{\hat{x}}{\hat{r}} \tag{21}$$

$$\frac{\partial d}{\partial\hat{y}}=\pm n_y-n_z\frac{\partial\hat{\alpha}}{\partial\hat{r}}\frac{\hat{y}}{\hat{r}}\,. \tag{22}$$

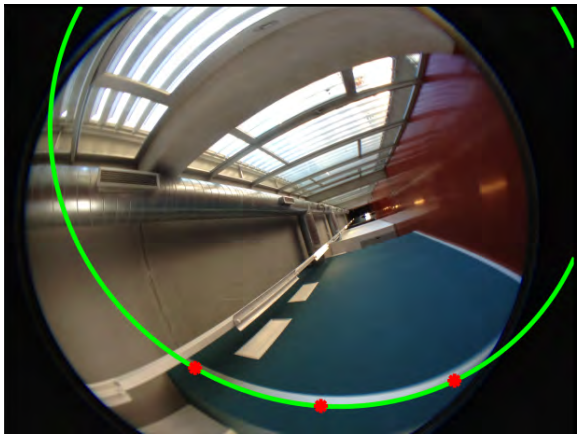
where $\frac{\partial\hat{\alpha}}{\partial\hat{r}}$ is computed from the definition of $\hat{\alpha}$ (see Table 4).

On the other hand, given an intensity image $I(x,y)$, the gradient $\nabla I(x,y)=(\nabla I_x,\nabla I_y)$ of each edge lying on a line-image is aligned with the gradient of the line-image distance (10). The dot product between the analytical gradient (22) and the vector $\nabla I^{\perp}=(-\nabla I_y,\nabla I_x)^T$ describes the parallelism between both gradients. This expression can be written as

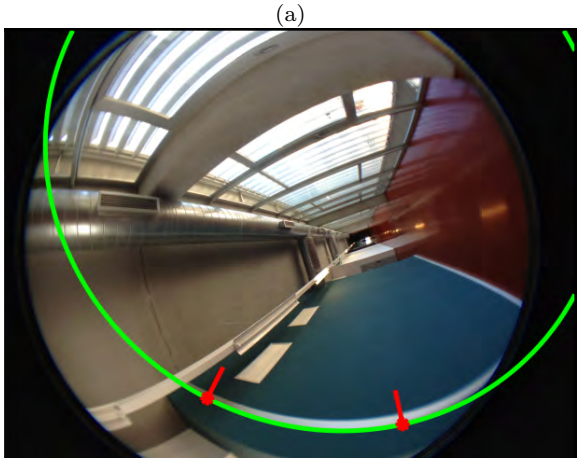
$$-\nabla I_y n_x\pm\nabla I_x n_y+n_z\frac{\partial\hat{\alpha}}{\partial\hat{r}}\frac{1}{\hat{r}}\left(\hat{x}\nabla I_y-\hat{y}\nabla I_x\right)=0, \tag{23}$$

and gives an additional constraint when solving a line-image. In particular, knowing the system calibration and given the location and the gradient of a single point lying on a line-image , the linear system

$$\begin{pmatrix}\hat{x} & \pm\hat{y} & -\hat{\alpha}\\ -\nabla I_y & \pm\nabla I_x & \frac{\partial\hat{\alpha}}{\partial\hat{r}}\frac{1}{\hat{r}}\left(\hat{x}\nabla I_y-\hat{y}\nabla I_x\right)\end{pmatrix}\mathbf{n}=\begin{pmatrix}0\\0\end{pmatrix} \tag{24}$$



(a)



(b)

Fig. 10 (a) Extraction of a line-image from three points.(b) Extraction of a line-image from two points using gradient constraint (in red the direction of ∇I). In this example, the linear approximation is used for the two points method.

can be solved for $\mathbf{n}$ resulting

$$\mathbf{n} = \begin{pmatrix} \nabla I_x \hat{\alpha} + \frac{\partial \hat{\alpha}}{\partial \hat{r}} \frac{\hat{y}}{\hat{r}} \left(\hat{x} \nabla I_y - \hat{y} \nabla I_x \right) \\ \nabla I_y \hat{\alpha} - \frac{\partial \hat{\alpha}}{\partial \hat{r}} \frac{\hat{x}}{\hat{r}} \left(\hat{x} \nabla I_y - \hat{y} \nabla I_x \right) \\ \hat{y} \nabla I_y + \hat{x} \nabla I_x \end{pmatrix} . \tag{25}$$

4.3 Unified Computation of Main Calibration Parameter $\hat{r}_{vl}$

In previous sections, we have solved the plumb-line problem using the exact expression of $\hat{\alpha}$. In this section we explore the alternative of approximating $\hat{\alpha}\left(\hat{r}\right)$ using a generic description depending on $\hat{r}$ and the main calibration parameter $\hat{r}_{vl}$. The goal of this approach is performing a direct method to exploit the gradient information in line-image extraction. In practice that means recovering the line-image and the main calibration parameter $\hat{r}_{vl}$ from two edge points reducing the number of iterations needed in a RANSAC approach.

Important aspects to consider are the degree of the approximation and the approximating point (the point around which the neighbourhood of the function is approximated).

Linear approximation When approximating $\hat{\alpha}\left(\hat{r}\right)$ by a linear function we consider the adequate approximating point is $\hat{r}_{vl}$. In this case $(\hat{r} = \hat{r}_{vl})$ implies $\hat{\alpha} = 0$ (see Section 3 and Fig. 4), and simplifying the expression it becomes

$$\hat{\alpha}\left(\hat{r}\right) = \frac{\partial \hat{\alpha}\left(\hat{r}_{vl}\right)}{\partial \hat{r}} \left(\hat{r} - \hat{r}_{vl}\right). \tag{26}$$

This kind of linear approximation is similar to series description of distortion function used in (Tardif et al, 2006). However, notice that in our approach we are approximating $\hat{\alpha}$ around $\hat{r} = \hat{r}_{vl}$ instead of approximating $\hat{\alpha}$ around $\hat{r} = 0$. When linearizing the function $\hat{\alpha}$ around $\hat{r} = 0$ the derivative of the function is equal to 0 (see Fig. 4) for all the systems meaning that the approximation is a constant $\hat{\alpha}\left(\hat{r}\right) = \hat{\alpha}\left(0\right)$ not depending on $\hat{r}$ which is a worse approximation [1].

In addition, linearising around $\hat{r}_{vl}$ allows computing $\hat{r}_{vl}$ even if the class of the central system is unknown. Using (26) as approximation of $\hat{\alpha}$ the system (24) becomes

$$\begin{pmatrix} \hat{x} & \pm\hat{y} & -\hat{r} & 1 \\ -\nabla I_y & \pm\nabla I_x & \frac{\hat{x}\nabla I_y - \hat{y}\nabla I_x}{\hat{r}} & 0 \end{pmatrix} \mathbf{m} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{27}$$

which can be solved from the coordinates and the gradient of two image points obtaining the vector $\mathbf{m} = \lambda\Big(n_x, n_y, n_z \frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}}, n_z \frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}} \hat{r}_{vl}\Big)^{\mathsf{T}}$. The main calibration parameter $\hat{r}_{vl}$ can be computed directly from $\mathbf{m}$ as $\hat{r}_{vl} = \frac{m_3}{m_4}$.

In this vector, n_z and $\frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}}$ are coupled, therefore $\mathbf{n}$ can not be completely recovered. However if we know the system kind we can include the value of $\frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}}$ in the equation system (see Table 5) and the equations system becomes

$$\begin{pmatrix} \hat{x} & \pm\hat{y} & -K_1\hat{r} & K_1 \\ -\nabla I_y & \pm\nabla I_x & K_1 \frac{\hat{x}\nabla I_y - \hat{y}\nabla I_x}{\hat{r}} & 0 \end{pmatrix} \mathbf{m} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{28}$$

where $\mathbf{m} = \lambda(n_x, n_y, n_z, n_z\hat{r}_{vl})^{\mathsf{T}}$. In this case both, $\hat{r}_{vl}$ and $\mathbf{n}$ are directly computed from $\mathbf{m}$.

Notice that in this case we have 3 DOFs and 4 equations, therefore we still have margin for using a higher approximation.

Second order approximation If we consider the second order approximation in the neighbourhood of $\hat{r} = \hat{r}_{vl}$

$$\hat{\alpha}\left(\hat{r}\right) = \frac{\partial \hat{\alpha}\left(\hat{r}_{vl}\right)}{\partial \hat{r}} \left(\hat{r} - \hat{r}_{vl}\right) + \frac{1}{2} \frac{\partial^2 \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}^2} \left(\hat{r} - \hat{r}_{vl}\right)^2 \tag{29}$$

which, using Table 5, can be expressed in terms of constants depending on the system kind

$$\hat{\alpha}\left(\hat{r}\right) = K_1 \left(\hat{r} - \hat{r}_{vl}\right) + \frac{1}{2} \frac{K_2}{\hat{r}_{vl}} \left(\hat{r} - \hat{r}_{vl}\right)^2 \tag{30}$$

therefore the system (24) becomes

$$\mathbf{m}^{\mathsf{T}} \begin{pmatrix} \hat{x} & -\nabla I_y \\ \pm\hat{y} & \pm\nabla I_x \\ -\hat{r}\left(K_1 - K_2\right) & \frac{(\hat{x}\nabla I_y - \hat{y}\nabla I_x)(K_1 - K_2)}{\hat{r}} \\ \left(K_1 - \frac{K_2}{2}\right) & 0 \\ -\frac{K_2}{2} r^2 & K_2\left(\hat{x}\nabla I_y - \hat{y}\nabla I_x\right) \end{pmatrix} = \begin{pmatrix} 0 & 0 \end{pmatrix} \tag{31}$$

where $\mathbf{m} = \lambda\big(n_x\hat{r}_{vl}, n_y\hat{r}_{vl}, n_z\hat{r}_{vl}, n_z\hat{r}_{vl}^2, n_z\big)^{\mathsf{T}}$. Having 4 DOFs this description allows to construct a system of four equations from two points lying on the line-image.

This result is particularly useful for the case of equiangular fisheye system. As presented in Section 4.1, most of the equations of Table 3 have an analytical solution. However in the case of the equiangular fisheye system the constraint has not a closed solution and it is solved by minimization.

[1] In the case of an orthogonal system the derivative of the function at $\hat{r} = \hat{r}_{vl}$ is ∞ meaning that this is not the proper point for linearisation

	Para-Catadioptric	Hyper-Catadioptric	Equiangular-Fisheye	Stereographic-Fisheye	Orthogonal-Fisheye	Equisolid-Fisheye
$K_0 = \hat{\alpha}(\hat{r}_{vl})$	0	0	0	0	0	0
$K_1 = \frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}}$	1	$\cos\chi$	$\frac{\pi}{2}$	1	∞	2
$K_2 = \frac{\partial^2 \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}^2}\hat{r}_{vl}$	1	$\cos^3\chi$	π	1	∞	6

Table 5 Derivatives of function $\hat{\alpha}(\hat{r})$ at $\hat{r}_{vl}$.

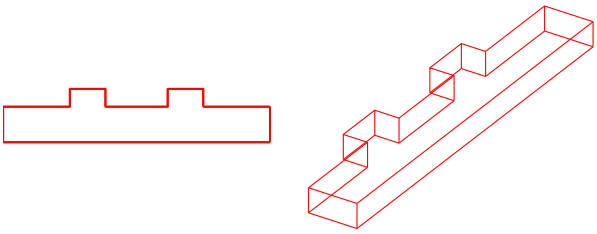


Fig. 11 (a) Top-view of the simulated scenario. (b) Isometric view of the simulated scenario.

In addition, in this last proposal for the case of equiangular-fisheye the value of $K_1 - \frac{K_2}{2} = 0$ meaning that the equations system (31) simplifies becoming

$$\mathbf{m}^{\mathsf{T}} \begin{pmatrix} \hat{x} & -\nabla I_y \\ \pm\hat{y} & \pm\nabla I_x \\ \frac{\pi}{2}\hat{r} & -\frac{\pi}{2\hat{r}}\left(\hat{x}\nabla I_y - \hat{y}\nabla I_x\right) \\ -\frac{\pi}{2}r^2 & \pi\left(\hat{x}\nabla I_y - \hat{y}\nabla I_x\right) \end{pmatrix} = \begin{pmatrix} 0 & 0 \end{pmatrix} \tag{32}$$

where $\mathbf{m} = \lambda(n_x\hat{r}_{vl}, n_y\hat{r}_{vl}, n_z\hat{r}_{vl}, n_z)^{\mathsf{T}}$ and the computing of $\mathbf{n}$ and $\hat{r}_{vl}$ is direct.

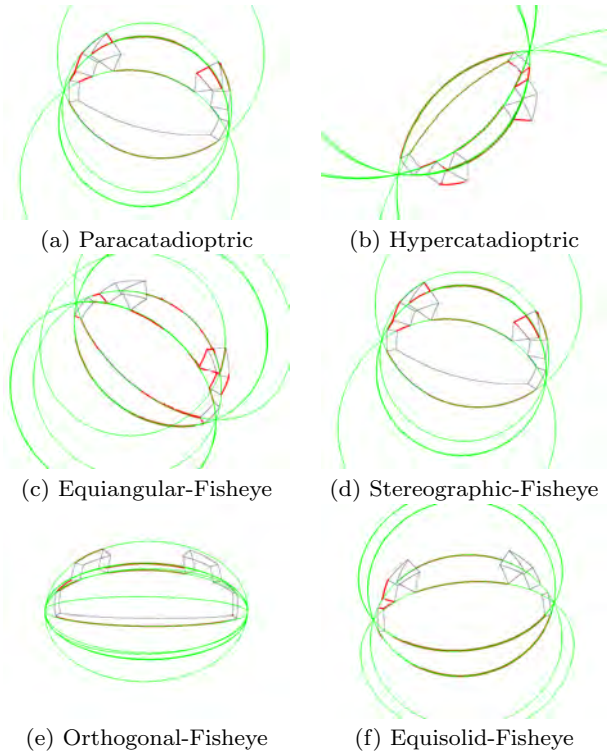
Finally, we consider the case of the orthogonal fisheye. As the derivative in $\hat{r}_{vl}$ is ∞ it makes sense to use a second order approximation in the neighbourhood of $\hat{r} = 0$ obtaining the system

$$\begin{pmatrix} \hat{x} & \pm\hat{y} & 1 & -\frac{1}{2}r^2 \\ -\nabla I_y & \pm\nabla I_x & 0 & \left(\hat{x}\nabla I_y - \hat{y}\nabla I_x\right) \end{pmatrix} \mathbf{m} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{33}$$

where $\mathbf{m} = \lambda\left(n_x\hat{r}_{vl}, n_y\hat{r}_{vl}, n_z\hat{r}_{vl}^2, n_z\right)^{\mathsf{T}}$.

In Fig. 10 (b) we show an example of a line-image defined with two points and the corresponding gradient of the intensity field I at these points. In this example, the linear approximation is used.

In all these methods we are using as input the coordinates of two points and the gradient orientation at these points. As gradient information is usually noisy, we propose low-pass filtering the gradient orientation to increase the precision.



(a) Paracatadioptric (b) Hypercatadioptric

(c) Equiangular-Fisheye (d) Stereographic-Fisheye

(e) Orthogonal-Fisheye (f) Equisolid-Fisheye

Fig. 12 Extraction example on synthetic images simulating different classes of central systems.

5 Uncalibrated Line-Image Extraction

In this section we present the algorithm for line-image extraction from uncalibrated omnidirectional cameras. The algorithm relies on the constraints presented in Section 4.1 based on a minimal set of 3 points. The algorithm is also developed for using gradient constraints (Section 4.3), based on a minimal set of 2 points. The proposed algorithm is described next (see Fig. 13).

The inputs of the method are the edges and their gradients obtained from a Canny detector (see Fig. 14 (a)). These edges are stored in connected components called boundaries and the gradient orientation is low-pass filtered to reduce noise. Then, a first splitting of the boundaries is done based on the variation of the gradient orientation. Each split is called a sub-boundary. The threshold of this splitting process has been cho-

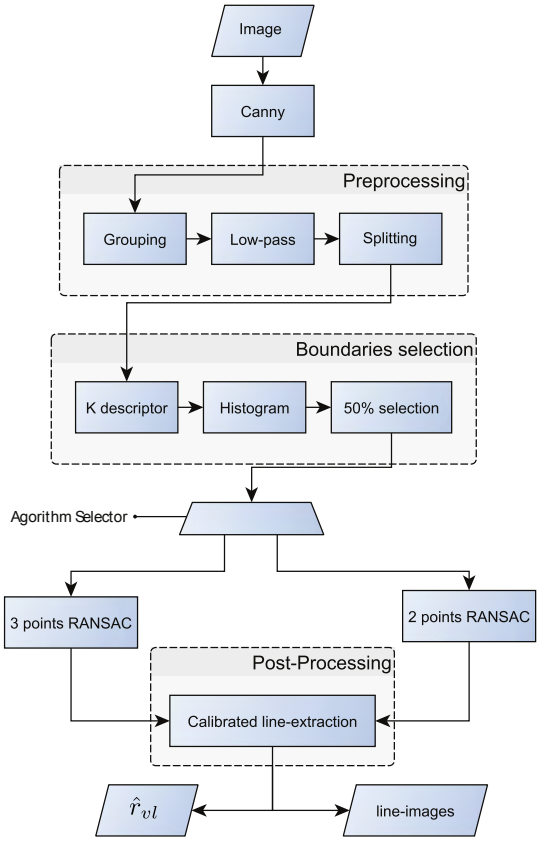


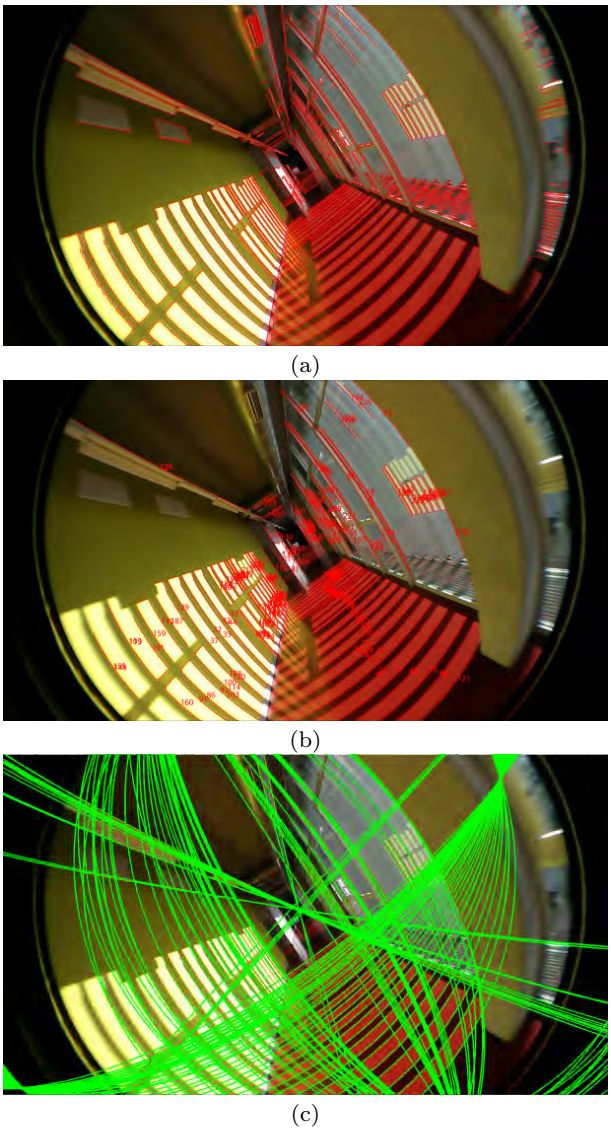
Fig. 13 Uncalibrated line-Image extraction procedure.

sen to minimize the number of splits. Therefore a sub-boundary can contain more than one segment. Not all the line-images contain relevant information about calibration (e.g. vertical lines). For this reason, a heuristic criterion is used to select a subset of these sub-boundaries containing relevant line-images. This heuristic technique is explained as follows.

We define the descriptor $\kappa = s_{size} \cdot \Delta\theta$ which increases as the sub-boundary is better for computing $\hat{r}_{vl}$ where s_{size} is the size of the sub-boundary and $\Delta\theta$ is the angle covered by the sub-boundary rounding the principal point. Sub-boundaries are sorted by κ and then the ones which cover 50% of an accumulated histogram of κ are chosen (see Fig. 14 (b)).

Once we have selected boundaries containing line-images well conditioned for computing $\hat{r}_{vl}$ we use a RANSAC-based approach (Fischler and Bolles, 1981) (see Algorithm 1). Points of a boundary are selected randomly to generate candidate line-images which are voted by the other points of the boundary. Two approaches are provided depending on the constraint considered:

3-points $\hat{r}_{vl}$ estimation: $\hat{r}_{vl}$ is estimated from the location coordinates of a minimum of three points of



(a) (b) (c)

Fig. 14 Steps in the line-image extraction: (a) Canny. (b) Preprocessing. (c) Line-Extraction

Algorithm 1 RANSAC approach for computing a set of line-images from a boundary of edge points.

for each boundary **do**
  while boundary contains line-images **do**
    for $iAttempt = 1$ to $nAttempts$ **do**
      $\hat{\mathbf{x}}_k = randomPoints\,(\hat{\mathbf{x}}_{boundary})$;
      $[\hat{r}_{vl}, \mathbf{n}] = computeLineImage\,(\hat{\mathbf{x}}_k)$
      $dist = computeDistance\,(\mathbf{n}, \hat{\mathbf{x}}_{boundary})$
      $\hat{\mathbf{x}}_{th} = computeThresholdPoint\,(\mathbf{n}, \hat{\mathbf{x}}_k, \delta_{px})$
      $threshold = computeDistance\,(\mathbf{n}, \hat{\mathbf{x}}_{th})$
      $votes\,(iAttempt) = dist < threshold$
    end for
    $lineImage.add\,(getMaxVotedLI\,(votes))$
    $\hat{\mathbf{x}}_{boundary} = removeVotingPoints\,(\hat{\mathbf{x}}_{boundary}, votes)$
  end while
end for

the boundary and the line-image constraint presented in Section 4.1.

2-points $\hat{r}_{vl}$ estimation: From the location coordinates of a minimum of two points and theirs gradients we can compute $\hat{r}_{vl}$ using the approximations presented in Section 4.3.

From any of these approaches, the normal vector $\mathbf{n}$ is computed from the defining points and $\hat{r}_{vl}$ using the expression (11). The distance to determine if a point is lying on the line-image is the algebraic distance (10) which is fast enough for the intensive evaluation needed in a RANSAC or in a Hough transform approach. In Fig. 8 we can see the different thickness for a given threshold depending on the different kind of systems. The region is not perfectly uniform along the curve but enough to decide which points belong to the line-image. Given a fixed threshold the thickness of this region also changes depending on the elevation of the projection plane $\mathbf{n}$. That means that a threshold being good enough for fitting some lines could not be good enough for others. To deal with this problem we compute a different threshold for each attempt of the RANSAC process (see Algorithm 1). Given a fixed threshold (δ_{px} in Algorithm 1) describing the thickness of the region in pixels, we compute the algebraic threshold corresponding to the line-image computed in the current attempt. As the algebraic threshold changes with each line-image we call it dynamic threshold (notice that δ_{px} remains constant). The process to estimate this dynamic threshold from δ_{px} and the vector $\mathbf{n}$ describing the line image is the following: Because of the nature of the RANSAC process we also have the location of the points defining the line-image of a given attempt ($\hat{\mathbf{x}}_k$ in Algorithm 1). As we have computed the line-image passing though these points we can compute the analytical gradient of the line-image on these points using equations (21-22) defining two straight lines passing though the points $\hat{\mathbf{x}}_k$ and orthogonal to the line-images (see Fig. 15). Along these straight lines we can define a point from a given distance δ_{px} of the line-image. On these points we compute the algebraic distance (10) and we consider its mean the threshold for the testing points.

Each point of the boundary whose distance is minor than the dynamic threshold gives a vote for the candidate. The most voted candidate is selected as best fit. From the previous procedure we obtain for each line-image: the points lying on it, the corresponding normal vector, and an estimation of the calibration via the $\hat{r}_{vl}$ value. From the collection of estimations for $\hat{r}_{vl}$ we compute a single value using the median. Thanks to the pre-selection based on κ most of the line-images return similar values of $\hat{r}_{vl}$ getting a small number of spurious. In Fig. 16 we show a box-plot representing

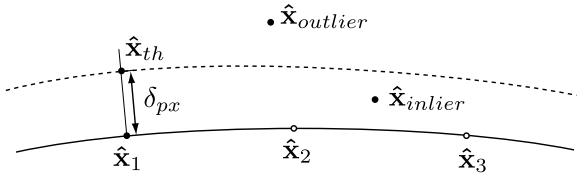


Fig. 15 Computing the dynamic threshold from the defining points of the line-image.

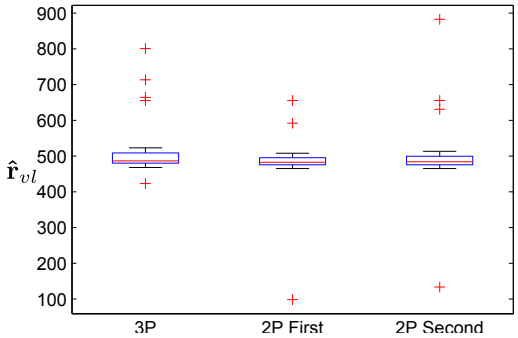


Fig. 16 Dispersion in values of $\hat{r}_{vl}$ for the different extracted line-images. 3P for 3-points approach, 2P First for 2-points linear approximation approach and 2P Second for 2-points second order approximation approach.

the dispersion in the values of $\hat{r}_{vl}$ from the line-images extracted in Fig. 14. The estimation for the value of $\hat{r}_{vl}$ is similar in the three approaches. Finally, main calibration parameter and normals are refined in a non-linear optimization process and then, the line-images are re-extracted using this value.

6 Experiments

We present different experiments to validate the proposal and to compare the two approaches presented in section 5. The accuracy in line extraction and calibration is measured using synthetic images with known ground-truth. We also evaluate the influence of the error in the principal point and compare our results with a calibration method. Finally, we present experiments to show the behaviour of the method with real images.

6.1 Experiments with Synthetic Data

To evaluate the accuracy of the algorithm we have used synthetic images generated for the different catadioptric and dioptric models considered in Section 2. A useful measure for comparing calibration estimations among the different systems is the radius of the vanishing line $\hat{r}_{vl}$. This radius depends only on the calibration and it is a magnitude common to any central system.

First experiment evaluates the deviation of the estimated radius of the vanishing line $\hat{r}_{vl}$. The setup of the

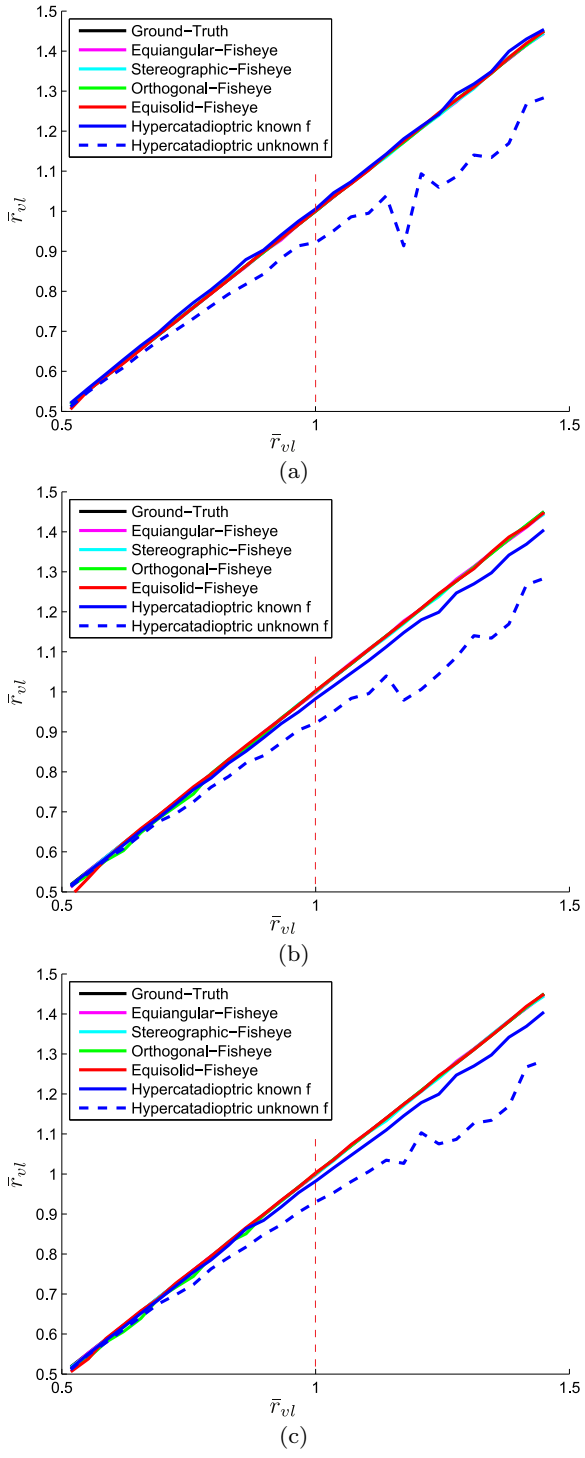


Fig. 17 Estimation of $\hat{r}_{vl}$ from uncalibrated line extraction. (a) 3-points approach, (b) 2-points linear approximation approach, (c) 2-points second order approximation approach. In all, the horizontal axis corresponds to the actual $\hat{r}_{vl}$ and the vertical axis to the estimated one.

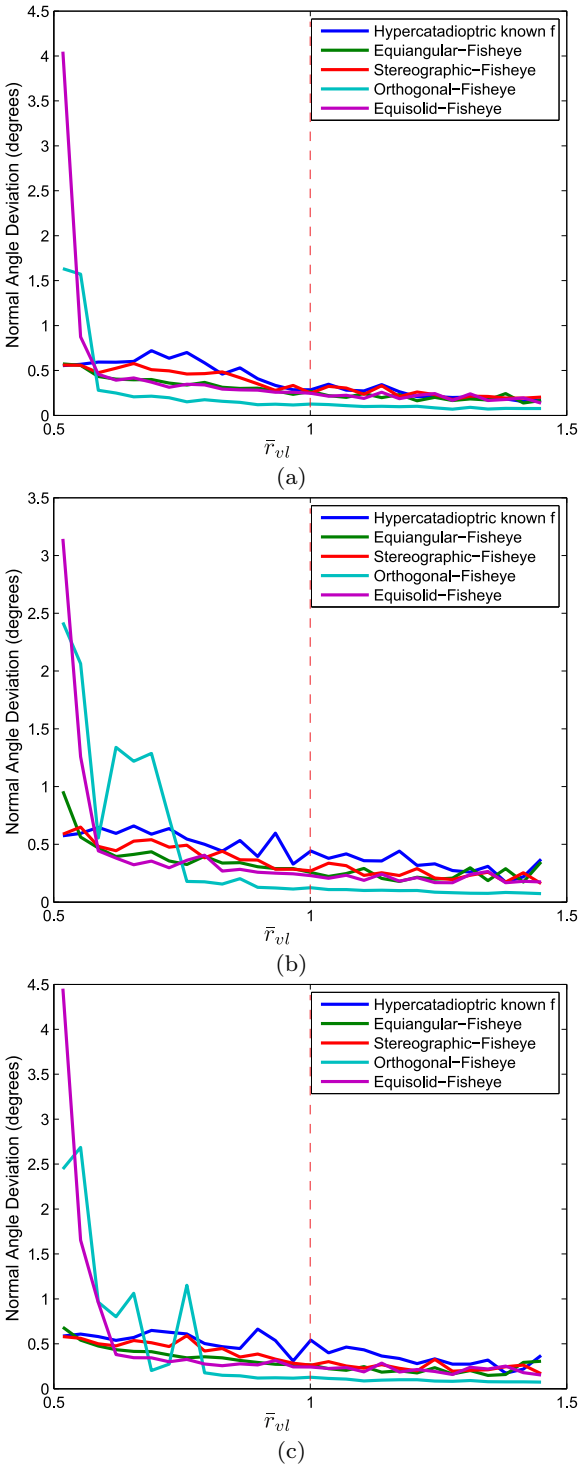


Fig. 18 Angle deviation between obtained normals and ground truth. (a) 3-points approach, (b) 2-points linear approximation approach, (c) 2-points second order approximation approach.

experiment is the following: Given an scene composed by segments in a corridor (see Fig. 11) we want to characterize the error of estimation in $\hat{r}_{vl}$. For each value of $\hat{r}_{vl}$ and for each system we generate 10 different images with a resolution of 1024×1024 pixels. A randomized perturbation in the orientation and location of the camera is introduced in each capture. From each image we obtain a value of $\hat{r}_{vl}$ and a set of line projections described by the normal vector $\mathbf{n}$. The experiment is repeated for each proposed method.

In Fig. 17 we show the deviation of the estimated radius of the vanishing line $\hat{r}_{vl}$, which has been normalized to make it independent from the image size. For each value of $\hat{r}_{vl}$ we have computed 10 estimations of $\hat{r}_{vl}$ using 10 different images; we represent the mean of these estimations. The normalized value $\bar{r}_{vl} = 1$ corresponds to the case of vanishing line projection passing through the corner of the image (e.g. with a resolution of 1024×1024, values in pixels of $\bar{r}_{vl} = \frac{\hat{r}_{vl}}{724}$). When $\bar{r}_{vl} < 1$ the FOV of the system is greater than 180 degrees. Otherwise the FOV of the system is smaller than 180 degrees.

Fig. 17 (a) depicts the results using the 3-points-$\hat{r}_{vl}$ estimation. Fig. 17 (b-c) depicts the results using the 2-points-$\hat{r}_{vl}$ estimation when using first and second approximation. For the case of the hypercatadioptric system we show two curves. The continuous curve corresponds to the estimation of $\hat{r}_{vl}$ assuming that parameter f is known. The dashed line corresponds to the the estimation of $\hat{r}_{vl}$ when none of the parameters are known.

Another measure of the accuracy in estimation is the deviation in the orientation of the normals $\mathbf{n}$ describing the line projections. In Fig. 18 we show the error in degrees between the ground truth normals and the estimated ones depending on $\bar{r}_{vl}$. From each image the mean of the deviations of all the line-projections is used. As we have 10 samples for each value of $\bar{r}_{vl}$ we also compute the median of these values.

We observe that, in general, accuracy in $\hat{r}_{vl}$ estimation decreases when the radius is greater than the image size (values of $\bar{r}_{vl}$ greater than one). That means that the smaller the FOV of the system the lesser the accuracy of the algorithm. The extreme case is the perspective camera where $\hat{r}_{vl} = \infty$ and curvature of lines is independent from the focal distance. We also note that the hypercatadioptric case is more difficult to solve than others because of the multiple parameters involved. The accuracy decreases considerably in systems with FOV less than 180 degrees. However if one of the calibration parameters is known the behaviour is similar to the other systems. In Fig. 12 we show some examples of the simulated images for different catadioptric and dioptric devices and the corresponding extracted line-images.

Influence of principal point accuracy In the previous experiment we have assumed that the principal point is known. To evaluate the robustness of the algorithm we evaluate the influence of Gaussian error in the principal point. The ground-truth of calibration $\hat{r}_{vl}$ is fixed to 750 pixels and the image size is 1024×768 pixel (FOV > 180 deg). In Fig. 19 we represent the error when estimating the main calibration parameter $\hat{r}_{vl}$ and the error in degrees between the ground-truth normals describing each line projection and the estimated ones. For the case of the hypercatadioptric system we estimate $\hat{r}_{vl}$ and the normals $\mathbf{n}$ assuming that parameter f is known. We can observe that the median of the error in the deviation of the normal does not exceed 0.5 degrees.

Calibration method comparison The objective of our proposal is the extraction of line projections in central system when the calibration is unknown. In addition we compute the radius of the vanishing line $\hat{r}_{vl}$ which is considered the main calibration parameter of the system. To illustrate the behaviour of our proposal we have carried out a "proof of concept" comparison between our approach and the method proposed in (Barreto and Araujo, 2005). We use the same imagery as input as in both methods.

Although the results from both methods can be directly compared due to the similarity of the projection models, there are some difference we need to take into account:

- The proposal of (Barreto and Araujo, 2005) is only valid for catadioptric systems, hence we cannot compare the approaches proposed for fisheye systems.
- In (Barreto and Araujo, 2005) the full calibration is performed in a closed form. Our proposal does not compute the principal point.
- Although a self-calibration is possible with (Barreto and Araujo, 2005) approach, in practice it is a hard problem with 5 degrees of freedom hence in the toolbox the user has to manually select the contours containing the line-images. By contrast our toolbox is automatic and the supervision of the user consists of the parameters tuning.

The input image is a synthetic image of a paracatadioptric system used in the previous experiments with a ground truth of $\hat{r}_{vl} = 500\ pixels$. The method presented in (Barreto and Araujo, 2005) gives a result of $\hat{r}_{vl} = 485.94\ pixels$. Our method, returns a value of $\hat{r}_{vl} = 500.36\ pixels$. Notice that, while Barreto's method is computing the full calibration, we as-

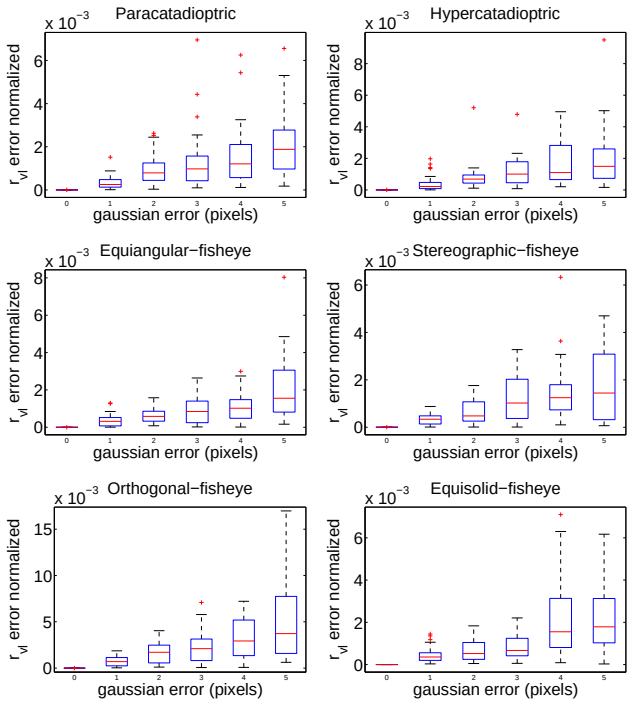
(a) (b) (c)

Fig. 20 Line extraction on uncalibrated omni-images (3-points algorithm). (a) Paracatadioptric. (b) Hypercatadioptric. (c) Equiangular-Fisheye

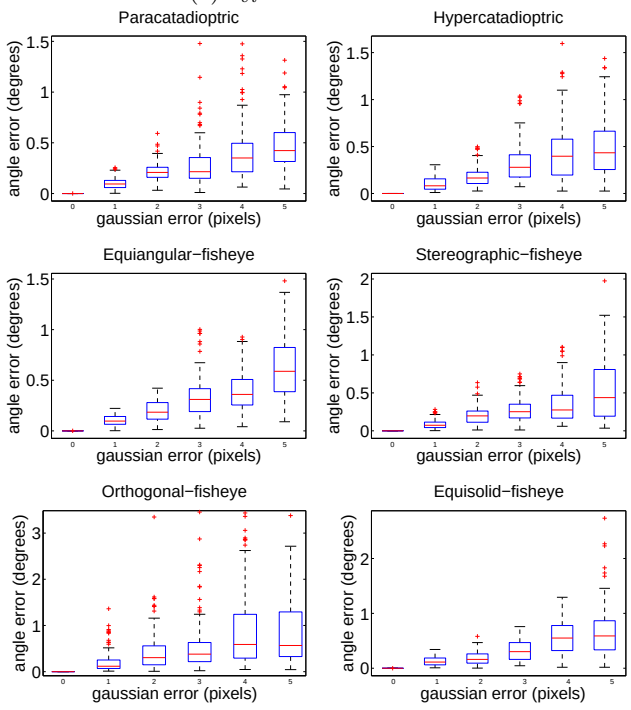
sume the principal point is known having not skewness. In Fig. 21 (a) we show the output image of the method presented in (Barreto and Araujo, 2005) containing the line-images used for calibrating and in Fig. 21 (b) the output of our proposal showing the extracted line-images. We can see that the parametric representation of the line-images in both images is similar.

6.2 Experiments with Real Images

In order to show how the method works with real images several tests have been performed with different catadioptric and fisheye cameras. The principal point has been coarsely estimated. The 3-points algorithm have been used for all these examples. In Fig. 20 (a) we show the behaviour of the proposal using paracatadioptric images from the CatPack toolbox (Barreto and Araujo, 2006). In Fig. 20 (b) we show our results in hy-

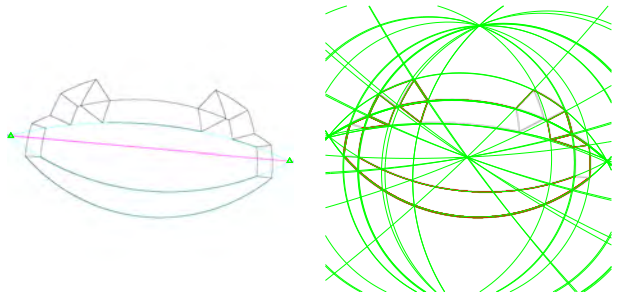


(a) $\hat{r}_{vl}$ estimation error.



(b) angle estimation error.

Fig. 19 Influence of principal point: (a) $\hat{r}_{vl}$ estimation error. (b) angle estimation error.



(a) (b)

Fig. 21 Output of the toolboxes showing the line-images: (a) Method of Barreto & Araujo. (b) Our proposal.

percatadioptric images with resolution of 1280 × 1024 pixels and has been acquired with a hypercatadioptric system composed of an IDS uEye camera and an hyperbolic mirror made by Neovision. In Fig. 20 (c) a fisheye based camera is used. The images has been taken with an iPhone 4S camera (3264×2498 pixels) and a Nexus 4 camera (3264×2448 pixels) with an equiangular fisheye for cell phones made by Pixeet. Notice that line-images of parallel 3D lines intersect in the vanishing points of the image because line-images have been well extracted.

In second experiment we have applied the algorithm to an image sequence taken with a camera in hand. The objective of this experiment is evaluating the robustness of the proposal when supervision is not available. The sequence has been acquired with a Nexus 4 camera using an equiangular fisheye with a resolution of 1920 × 1080 pixels [2]. Each frame is independently computed to emphasize the robustness of the proposal without supervision. The processing has been off-line computed using the 3-points approach. In Fig. 22(b) we show the estimation of the main calibration parameter $\hat{r}_{vl}$ along the sequence. The mean estimation for $\hat{r}_{vl}$ is 568.41 pixels and the standard deviation is 9 pixels, meaning that the proposal is robust enough to be applied in a sequence of images (an implementation in real-time must consider aspects not addressed in this paper). As this scenario is composed of parallel lines, a qualitative measure of the quality of the extracted line-images is the location of the vanishing points on these images. Line-images from video (see Fig. 22(a)) intersect at the vanishing points of the image (notice that we do not have imposed this constraint).

The presented approach can be tested with a toolbox for Matlab [3] available on line. This toolbox allows to obtain the calibration of a fisheye or a catadioptric system just from a single picture. The setup is very sim-

[2] webdiis.unizar.es/%7Ebermudez/suppMat.html
[3] webdiis.unizar.es/%7Ebermudez/toolbox.html

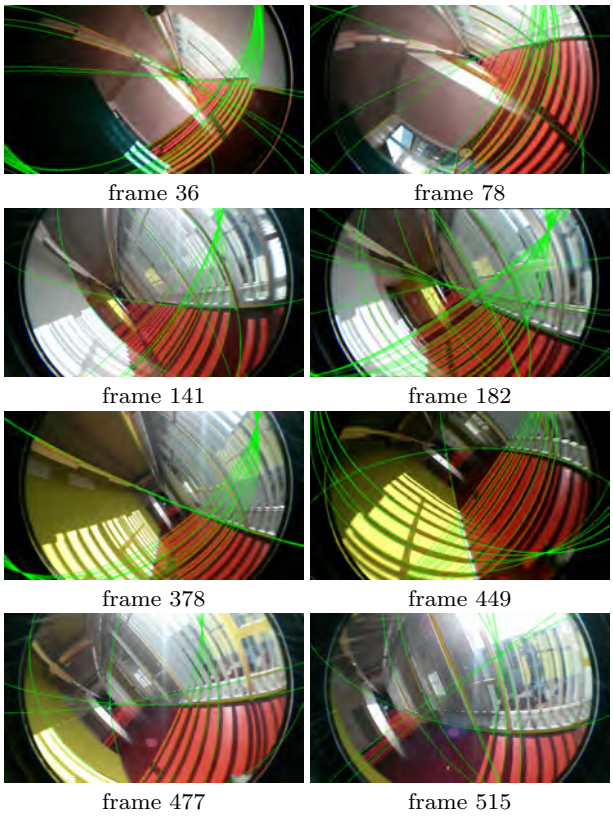


Fig. 22 Line extraction on uncalibrated sequence: (a) Extracted line-images in sequence (b) $\hat{r}_{vl}$ estimation in blue) $\hat{r}_{vl}$ mean value in red

ple because neither specific pattern nor special movements of the camera are needed.

7 Conclusion

We have presented a method to extract line-images from uncalibrated central images with revolution symmetry. We consider a general framework which encodes projection models for dioptric and catadioptric system using a common main calibration parameter (the radius of the vanishing line $\hat{r}_{vl}$). We characterize the line projections for different types of catadioptric and dioptric systems evaluating common properties. Due to the relation between projection models and curvature of line-images an estimation of the main calibration parameter of the system is obtained simultaneously. We observed that line-images are correctly extracted and the obtained normals are accurate enough to be used in 3D computations. Nevertheless, in hypercatadioptric systems the accuracy is lower because two calibration parameters are involved. However, the accuracy is similar to the other systems if one of the parameters is known or when the FOV of the system is greater than 180 degrees. Besides characterizing the proposal it has been tested with real images from catadioptric and dioptric systems. Tests with real and synthetic images can be replicated with an implementation of the method provided as open source for research purposes.

Acknowledgements This work was supported by the Spanish project VINEA DPI2012-31781 and FEDER funds. First author was supported by the FPU program AP2010-3849. Thanks to J. P. Barreto from ISR Coimbra for the set of high resolution paracatadioptric images. The final publication is available at Springer via http://dx.doi.org/10.1007/s11263-014-0792-7.

A Polynomials Describing Line-Images

Some of the line-images in central systems with revolution symmetry can be expressed as polynomials. In this Appendix we show the description of these line images using polynomials for Catadioptric systems, Equisolid-fisheye, Stereographic-fisheye, Orthogonal-fisheye and Equisolid-fisheye systems.

Catadioptric and Stereographic-fisheye

$$\left(\hat{x}^2\ \hat{x}\hat{y}\ \hat{y}^2\ \hat{x}\ \hat{y}\ 1\right)\boldsymbol{\Omega}_{cata}=0 \tag{34}$$

where

$$\boldsymbol{\Omega}_{cata}=\begin{pmatrix} n_x^2\sin\chi^2-n_z^2\cos\chi^2 \\ 2n_xn_y\sin\chi^2 \\ n_y^2\sin\chi^2-n_z^2\cos\chi^2 \\ 2f\sin\chi n_xn_z \\ 2f\sin\chi n_yn_z \\ f^2\sin\chi^2n_z^2 \end{pmatrix} \tag{35}$$

Ortogonal-Fisheye

$$\left(\hat{x}^2\ \hat{x}\hat{y}\ \hat{y}^2\ \hat{x}\ \hat{y}\ 1\right)\boldsymbol{\Omega}_{ortho}=0 \tag{36}$$

where

$$\boldsymbol{\Omega}_{ortho}=\begin{pmatrix} -n_x^2-n_z^2 \\ -2n_xn_y \\ -n_y^2-n_z^2 \\ 0 \\ 0 \\ f^2n_z^2 \end{pmatrix} \tag{37}$$

Equisolid-Fisheye

$$\left(\hat{x}^4\ \hat{x}^3\hat{y}\ \hat{x}^2\hat{y}^2\ \hat{x}^2\ \hat{x}\hat{y}^3\ \hat{x}\hat{y}\ \hat{y}^4\ \hat{y}^2\ 1\right)\mathbf{Q}=0 \tag{38}$$

where

$$\mathbf{Q}=\begin{pmatrix} -4\left(n_x^2+n_z^2\right) \\ -8n_xn_y \\ -4n_x^2-4n_y^2-8n_z^2 \\ 4f^2\left(n_x^2+n_z^2\right) \\ -8n_xn_y \\ 8f^2n_xn_y \\ -4\left(n_y^2+n_z^2\right) \\ 4f^2\left(n_y^2+n_z^2\right) \\ -f^4n_z^2 \end{pmatrix} \tag{39}$$

B Computing the Focal Distance in Hypercatadioptric Systems

In this Appendix we expand a way for computing the focal length in hypercatadioptic systems. This allows us to compute both calibration parameters. Instead of using the plumb-line constraint or a combination between plum-line and gradient constraints the normal $\mathbf{n}$ can be computed from a pair of points using the gradient constraint (23).

$$\begin{pmatrix} -\nabla I_{y1} & \pm\nabla I_{x1} & \frac{\partial\hat{\alpha}_1}{\partial\hat{r}}\frac{1}{\hat{r}_1}\left(\hat{x}_1\nabla I_{y1}-\hat{y}_1\nabla I_{x1}\right) \\ -\nabla I_{y2} & \pm\nabla I_{x2} & \frac{\partial\hat{\alpha}_2}{\partial\hat{r}}\frac{1}{\hat{r}_2}\left(\hat{x}_2\nabla I_{y2}-\hat{y}_2\nabla I_{x2}\right) \end{pmatrix}\mathbf{n}=\begin{pmatrix}0\\0\end{pmatrix} \tag{40}$$

In practice, the solution is noisy and does not imply a real advantage with respect to the two point's location approach (11).

The previous constraint (40) using 2 points and their gradients is enough to define a line-image. Therefore, when adding a third point with (23) in (40) one of the equations can be expressed in combination of the other two. In practice this means that,

$$\varpi_1\frac{\partial\hat{\alpha}_1}{\partial\hat{r}}\frac{1}{\hat{r}_1}+\varpi_2\frac{\partial\hat{\alpha}_2}{\partial\hat{r}}\frac{1}{\hat{r}_2}+\varpi_3\frac{\partial\hat{\alpha}_3}{\partial\hat{r}}\frac{1}{\hat{r}_3}=0 \tag{41}$$

where

$$\varpi_1=\left(\nabla I_{x2}\nabla I_{y3}-\nabla I_{x3}\nabla I_{y2}\right)\left(\hat{x}_1\nabla I_{y1}-\hat{y}_1\nabla I_{x1}\right) \tag{42}$$
$$\varpi_2=\left(\nabla I_{x3}\nabla I_{y1}-\nabla I_{x1}\nabla I_{y3}\right)\left(\hat{x}_2\nabla I_{y2}-\hat{y}_2\nabla I_{x2}\right) \tag{43}$$
$$\varpi_3=\left(\nabla I_{x1}\nabla I_{y2}-\nabla I_{x2}\nabla I_{y1}\right)\left(\hat{x}_3\nabla I_{y3}-\hat{y}_3\nabla I_{x3}\right) \tag{44}$$

This constraint is similar to (13) but using gradients (notice that location information is also used).

As noted before, gradient information is noisier than location information, therefore there is no advantage in using this constraint instead of (13). However, there is a case in which this constraint is useful. The constraint (13) is solved for each system in Table 3. Most of the devices taken into account have a single calibration parameter defining distortion. This is the case of equiangular, stereographic, orthogonal and equisolid. The parabolic case is defined by two parameters f and p but a coupled parameter $r_{vl}=fp$ can be used instead. In these cases the constraint involving three points can be used to estimate the calibration of the system. However, in the hyperbolic case the two parameters χ and f can not be coupled and only one of them can be estimated from this constraint.

By contrast, when simplifying the equation (41) for hypercatadioptric systems we found that mirror parameter χ is not involved:

$$\varpi_1\frac{1}{\sqrt{\hat{r}_1^2+f^2}}+\varpi_2\frac{1}{\sqrt{\hat{r}_2^2+f^2}}+\varpi_3\frac{1}{\sqrt{\hat{r}_3^2+f^2}}=0 \tag{45}$$

As consequence, the focal distance f can be computed from the gradient orientation and the location of three points lying on a line-image.

Equation 45 can be expressed as a polynomial of degree 8 (but bi-quartic),

$$\sum_{m=0}^{4}\beta_m f^{2m}=0 \tag{46}$$

where $\beta_m=\beta_{m,123}+\beta_{m,213}+\beta_{m,312}$ and

$$\beta_{0,ijk}=-\varpi_i^4\hat{r}_j^4\hat{r}_k^4+2\hat{r}_i^4\hat{r}_j^2\hat{r}_k^2\varpi_j^2\varpi_k^2 \tag{47}$$

$$\beta_{1,ijk}=2\left(2\hat{r}_i^2\hat{r}_j^2\hat{r}_k^2+\hat{r}_i^4\left(\varpi_j^2+\varpi_k^2\right)\right)\varpi_j^2\varpi_k^2+\ldots -2\varpi_i^4\hat{r}_j^2\hat{r}_k^2\left(\hat{r}_j^2+\hat{r}_k^2\right) \tag{48}$$

$$\beta_{2,ijk}=\hat{r}_j^2\hat{r}_k^2\left(4\varpi^2\left(\varpi_j^2+\varpi_k^2-\varpi_i^2\right)+2\varpi_j^2\varpi_k^2\right)\ldots -\hat{r}_i^4\left(\varpi_j^2-\varpi_k^2\right)^2 \tag{49}$$

$$\beta_{3,ijk}=\hat{r}_i^2\left(\varpi_i^2\left(\varpi_j^2+\varpi_k^2\right)+2\varpi_j^2\varpi_k^2-\left(\varpi_j^4+\varpi_k^4\right)\right) \tag{50}$$

$$\beta_{4,ijk}=-\varpi_i^4+2\varpi_j^2\varpi_k^2 \tag{51}$$

This equation has 4 solutions (because the negative values for f have not sense).

C Coefficients for the Equisolid-Fisheye Plumb-Line Equation

In this Appendix we present the coefficient of the 16th degree polynomial to solve the equisolid plumb-line equation (19).

$$\sum_{m=0}^{8}\omega_m\hat{r}_{vl}^{2m}=0 \tag{52}$$

where $\omega_m=\omega_{m,123}+\omega_{m,213}+\omega_{m,312}$ and

$$\omega_{0,ijk}=-l_i^4\hat{r}_i^8\hat{r}_j^4\hat{r}_k^4+2\hat{r}_i^4l_j^2l_k^2\hat{r}_j^6\hat{r}_k^6 \tag{53}$$

$$\omega_{1,ijk}=4\left(\hat{r}_i^2\left(\hat{r}_j^2+\hat{r}_k^2\right)+\hat{r}_j^2\hat{r}_k^2\right)\left(l_i^4\hat{r}_i^6\hat{r}_k^2-2l_j^2l_k^2\hat{r}_i^2\hat{r}_j^4\hat{r}_k^4\right) \tag{54}$$

$$\begin{aligned}\omega_{2,ijk} = &-16l_i^4\hat{r}_i^4\left(\hat{r}_i^2\hat{r}_j^2\hat{r}_k^2\left(\hat{r}_j^2+\hat{r}_k^2\right)+\hat{r}_i^4\hat{r}_j^2\hat{r}_k^2\right)+...\\ &+32l_j^2l_k^2\hat{r}_j^2\hat{r}_k^2\left(\hat{r}_i^2\hat{r}_j^2\hat{r}_k^2\left(\hat{r}_j^2+\hat{r}_k^2\right)+\hat{r}_i^4\hat{r}_j^2\hat{r}_k^2\right)+...\\ &-2l_i^4\hat{r}_i^4\left(2\hat{r}_i^4\left(\hat{r}_j^4+\hat{r}_k^4\right)+3\hat{r}_j^4\hat{r}_k^4\right)+...\\ &+10l_j^2l_k^2\hat{r}_j^2\hat{r}_k^2\hat{r}_i^2\left(\hat{r}_j^4+\hat{r}_k^4\right)+...\\ &8l_j^2l_k^2\hat{r}_j^6\hat{r}_k^6\end{aligned} \quad (55)$$

$$\begin{aligned}\omega_{3,ijk} = &16l_i^4\hat{r}_i^6\left(\hat{r}_i^2\hat{r}_j^2+\hat{r}_i^2\hat{r}_k^2+\hat{r}_j^4+\hat{r}_k^4+4\hat{r}_j^2\hat{r}_k^2\right)+...\\ &-8l_i^2\hat{r}_i^6\left(4\left(l_j^2\hat{r}_j^4+l_k^2\hat{r}_k^2\right)+5\hat{r}_j^2\hat{r}_k^2\left(l_j^2+l_k^2\right)\right)+...\\ &-8\hat{r}_i^2\hat{r}_j^4\hat{r}_k^4\left(5l_i^2\left(l_j^2+l_k^2\right)+16l_j^2l_j^2\right)+...\\ &+4l_i^4\left(6\hat{r}_i^4\hat{r}_j^2\hat{r}_k^2\left(\hat{r}_j^2+\hat{r}_k^2\right)+\hat{r}_i^2\hat{r}_j^4\hat{r}_k^4\right)+...\\ &-4l_i^2\hat{r}_i^6\left(l_j^2\hat{r}_k^4+l_k^2\hat{r}_j^4\right)\end{aligned} \quad (56)$$

$$\begin{aligned}\omega_{4,ijk} = &+128l_j^2l_k^2\hat{r}_j^4\hat{r}_k^4+160\hat{r}_i^4l_i^2\hat{r}_j^2\hat{r}_k^2\left(l_j^2+l_k^2\right)+...\\ &-l_i^4\left(\hat{r}_j^4\hat{r}_k^4+64\hat{r}_i^6\left(\hat{r}_j^2+\hat{r}_k^2\right)\right)+...\\ &-16l_i^4\left(\hat{r}_i^8+\hat{r}_i^2\hat{r}_j^2\hat{r}_k^2\left(\hat{r}_j^2+\hat{r}_k^2\right)\right)+...\\ &+4\hat{r}_i^6l_i^2\left(4\left(l_j^2\hat{r}_k^2+l_k^2\hat{r}_j^2\right)+10\left(l_j^2\hat{r}_j+l_k^2\hat{r}_k^2\right)\right)+...\\ &\hat{r}_i^4l_j^2l_k^2\left(16\left(\hat{r}_j^2+\hat{r}_k^2\right)+50\hat{r}_j^2\hat{r}_k^2\right)+...\\ &-24l_i^4\hat{r}_i^4\left(\left(\hat{r}_j^4+\hat{r}_k^4\right)+4\hat{r}_j^2\hat{r}_k^2\right)\end{aligned} \quad (57)$$

$$\begin{aligned}\omega_{5,ijk} = &64l_i^2\hat{r}_i^2\left(l_i^2\hat{r}_i^4+l_i^2\hat{r}_j^2\hat{r}_k^2-\hat{r}_i^2\left(l_j^2\hat{r}_k^2+l_k^2\hat{r}_j^2\right)\right)+...\\ &+32l_i^2\hat{r}_i^2\left(3l_i^2\hat{r}_i^2\left(\hat{r}_j^2+\hat{r}_k^2\right)-5\hat{r}_i^2\left(l_j^2\hat{r}_j^2+l_k^2\hat{r}_k^2\right)\right)+...\\ &-200l_j^2l_k^2\hat{r}_i^2\hat{r}_j^2\hat{r}_k^2+...\\ &16l_i^2\hat{r}_i^2\left(l_i^2\left(\hat{r}_j^4+\hat{r}_k^4\right)-\hat{r}_i^4\left(l_j^2+l_k^2\right)\right)+...\\ &+\left(\hat{r}_j^2+\hat{r}_k^2\right)\left(4l_i^4\hat{r}_j^2\hat{r}_k^2-20\hat{r}_i^4l_j^2l_k^2\right)\end{aligned} \quad (58)$$

$$\begin{aligned}\omega_{6,ijk} = &64l_i^2\hat{r}_i^2\left(-l_i^2\left(\hat{r}_j^2+\hat{r}_k^2\right)+\hat{r}_i^2\left(l_j^2+l_k^2\right)\right)+...\\ &16l_i^2\hat{r}_i^2\left(-6l_i^2\hat{r}_i^2+5\left(l_j^2\hat{r}_k^2+l_k^2\hat{r}_j^2\right)\right)+...\\ &+4\left(50l_j^2\hat{r}_j^2l_k^2\hat{r}_k^2-4l_i^4\hat{r}_j^2\hat{r}_k^2-l_i^4\left(\hat{r}_j^4+\hat{r}_k^4\right)+2\hat{r}_i^4l_j^2l_k^2\right)\end{aligned} \quad (59)$$

$$\omega_{7,ijk} = 16\left(4l_i^4\hat{r}_i^2+l_i^4\hat{r}_j^2+l_i^4\hat{r}_k^2-2\hat{r}_i^2l_j^2l_k^2-5l_i^2\hat{r}_i^2\left(l_j^2+l_k^2\right)\right) \quad (60)$$

$$\omega_{8,ijk} = 16\left(-l_i^4+2l_j^2l_k^2\right) \quad (61)$$

References

Alemán-Flores M, Alvarez L, Gomez L, Santana-Cedrés D (2014) Line detection in images showing significant lens distortion and application to distortion correction. Pattern Recognition Letters 36:261–271

Alvarez L, Gómez L, Sendra J (2009) An algebraic approach to lens distortion by line rectification. Journal of Mathematical Imaging and Vision 35(1):36–50

Baker S, Nayar SK (1999) A theory of single-viewpoint catadioptric image formation. International Journal of Computer Vision 35(2):175–196

Barreto JP, Araujo H (2005) Geometric properties of central catadioptric line images and their application in calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8):1327–1333

Barreto JP, Araujo H (2006) Fitting conics to paracatadioptric projections of lines. Computer Vision and Image Understanding 101(3):151–165

Bazin JC, Demonceaux C, Vasseur P, Kweon I (2010) Motion estimation by decoupling rotation and translation in catadioptric vision. Computer Vision and Image Understanding 114(2):254–273

Bermudez-Cameo J, Lopez-Nicolas G, Guerrero JJ (2012a) A unified framework for line extraction in dioptric and catadioptric cameras. In: 11th Asian Conference on Computer Vision, (ACCV), vol 7727

Bermudez-Cameo J, Puig L, Guerrero JJ (2012b) Hypercatadioptric line images for 3D orientation and image rectification. Robotics and Autonomous Systems 60(6):755–768

Bermudez-Cameo J, Lopez-Nicolas G, Guerrero JJ (2013) Line extraction in uncalibrated central images with revolution symmetry. In: 24th British Machine Vision Conference (BMVC)

Brown D (1971) Close-range camera calibration. Photogrammetric engineering 37(8):855–866

Bukhari F, Dailey MN (2013) Automatic radial distortion estimation from a single image. Journal of Mathematical Imaging and Vision 45(1):31–45

Courbon J, Mezouar Y, Eck L, Martinet P (2007) A generic fisheye camera model for robotic applications. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 1683–1688

Cucchiara R, Grana C, Prati A, Vezzani R (2003) A hough transform-based method for radial lens distortion correction. In: 12th International Conference on Image Analysis and Processing (ICIAP), pp 182–187

Devernay F, Faugeras O (2001) Straight lines have to be straight. Machine Vision and Applications 13(1):14–24

Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6):381–395

Fitzgibbon AW (2001) Simultaneous linear estimation of multiple view geometry and lens distortion. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol 1, pp I–125

Gasparini S, Caglioti V (2011) Line localization from single catadioptric images. International journal of computer vision 94(3):361–374

Gasparini S, Sturm P, Barreto JP (2009) Plane-based calibration of central catadioptric cameras. In: IEEE 12th International Conference on Computer Vision (ICCV), IEEE, pp 1195–1202

Geyer C, Daniilidis K (2000) A unifying theory for central panoramic systems and practical applications. In: 6th European Conference on Computer Vision, (ECCV), vol 2, pp 445–461

Geyer C, Daniilidis K (2001) Catadioptric projective geometry. International Journal of Computer Vision 45(3):223–243

Kannala J, Brandt S (2006) A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(8):1335–1340

Kannala J, Brandt SS, Heikkilä J (2008a) Self-calibration of central cameras by minimizing angular error. In: 3rd International Conference on Computer Vision Theory and Applications (VISAPP), pp 28–35

Kannala J, Heikkilä J, Brandt SS (2008b) Geometric camera calibration. Wiley Encyclopedia of Computer Science and Engineering

Kingslake R (1989) A history of the photographic lens. Academic Press

Mei C, Rives P (2007) Single viewpoint omnidirectional camera calibration from planar grids. In: International Conference on Robotics and Automation (ICRA), pp 3945–3950

Melo R, Antunes M, Barreto J, Falco G, Gonalves N (2013) Unsupervised intrinsic calibration from a single frame using a "plumb-line" approach. In: IEEE 14th International Conference on Computer Vision (ICCV), pp 1–6

Puig L, Bastanlar Y, Sturm P, Guerrero JJ, Barreto J (2011) Calibration of central catadioptric cameras using a dlt-like approach. International Journal of Computer Vision 93(1):101–114

Puig L, Bermudez-Cameo J, Sturm P, Guerrero JJ (2012) Calibration of omnidirectional cameras in practice. a comparison of methods. Computer Vision and Image Understanding 116:120–137

Ray S (2002) Applied photographic optics: Lenses and optical systems for photography, film, video, electronic and digital imaging. Focal Press

Rosten E, Loveland R (2011) Camera distortion self-calibration using the plumb-line constraint and minimal hough entropy. Machine Vision and Applications 22(1):77–85

Scaramuzza D, Martinelli A, Siegwart R (2006) A toolbox for easily calibrating omnidirectional cameras. In: International Conference on Ingelligent Robots and Systems (IROS)

Schneider D, Schwalbe E, Maas HG (2009) Validation of geometric models for fisheye lenses. Journal of Photogrammetry and Remote Sensing 64(3):259 – 266

Stevenson D, Fleck M (1996) Nonparametric correction of distortion. In: 3rd IEEE Workshop on Applications of Computer Vision (WACV), pp 214–219

Strand R, Hayman E (2005) Correcting radial distortion by circle fitting. In: 16th British Machine Vision Conference (BMVC)

Sturm P, Ramalingam S, Tardif JP, Gasparini S, Barreto JP (2011) Camera models and fundamental concepts used in geometric computer vision. Foundations and Trends in Computer Graphics and Vision 6(1-2):1–183

Swaminathan R, Nayar SK (2000) Nonmetric calibration of wide-angle lenses and polycameras. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10):1172–1178

Tardif J, Sturm P, Roy S (2006) Self-calibration of a general radially symmetric distortion model. 9th European Conference on Computer Vision (ECCV) pp 186–199

Thormählen T, Broszio H, Wassermann I (2003) Robust line-based calibration of lens distortion from a single view. In: Proceedings of MIRAGE, pp 105–112

Wang A, Qiu T, Shao L (2009) A simple method of radial distortion correction with centre of distortion estimation. Journal of Mathematical Imaging and Vision 35(3):165–172

Wu Y, Hu Z (2005) Geometric invariants and applications under catadioptric camera model. In: 10th IEEE International Conference on Computer Vision (ICCV), vol 2, pp 1547–1554

Ying X, Hu Z (2004a) Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model? In: 8th European Conference on Computer Vision (ECCV)

Ying X, Hu Z (2004b) Catadioptric line features detection using hough transform. In: 17th International Conference on Pattern Recognition (ICPR), vol 4, pp 839–842

Ying X, Zha H (2005) Simultaneously calibrating catadioptric camera and detecting line features using hough transform. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 412–417